Volume 5

Pages 924 - 1128

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

BEFORE THE HONORABLE WILLIAM H. ALSUP

ORACLE AMERICA, INC., Plaintiff, VS. GOOGLE, INC., Defendant.	No. C 10-3561 WHA
	San Francisco, California Friday, May 13, 2016

TRANSCRIPT OF PROCEEDINGS

APPEARANCES:

For Plaintiff:

ORRICK, HERRINGTON & SUTCLIFFE LLP
The Orrick Building
405 Howard Street
San Francisco, California 94105
BY: ANNETTE L. HURST, ESQUIRE
GABRIEL M. RAMSEY, ESQUIRE

(Appearances continued on next page)

Reported By: Pamela A. Batalo, CSR No. 3593, RMR, FCRR
Official Reporters - U.S. District Court

APPEARANCES (CONTINUED):

For Plaintiff: ORRICK, HERRINGTON & SUTCLIFFE LLP
The Orrick Building
51 West 52nd Street
New York, New York 10019
BY: PETER A. BICKS, ESQUIRE
LISA T. SIMPSON, ESQUIRE

ORACLE
500 Oracle Parkway
Redwood City, California 94065
BY: DORIAN ESTELLE DALEY, GENERAL COUNSEL
MATTHEW SARBORARIA,VICE PRESIDENT
ASSOCIATE GENERAL COUNSEL

For Defendant: KEKER & VAN NEST
633 Battery Street
San Francisco, California 94111-1809
BY: ROBERT A. VAN NEST, ESQUIRE
CHRISTA M. ANDERSON, ESQUIRE
MICHAEL S. KWUN, ESQUIRE
DANIEL PURCELL, ESQUIRE
MATTHIAS ANDREAS KAMBER, ESQUIRE
EUGENE MORRIS PAIGE, ESQUIRE
STEVEN P. RAGLAND, ESQUIRE
MATTHIAS A. KAMBER, ESQUIRE

KING & SPALDING LLP
1185 Avenue of the Americas
New York, New York 10036-4003
BY: BRUCE W. BABER, ESQUIRE

GOOGLE, INC.
1600 Amphitheatre Parkway
Mountain View, California 94043
BY: RENNY HWANG, LITIGATION COUNSEL

Also Present:

GEORGES SAAB
ORACLE CORPORATE REPRESENTATIVE

CATHERINE LACAVERA
GOOGLE CORPORATE REPRESENTATIVE

I N D E X

Friday, May 13, 2016 - Volume 5

E X H I B I T S

I N D E X

E X H I B I T S

Friday - May 13, 2016 **7:24 a.m.**

P R O C E E D I N G S

---000---

(Proceedings were heard out of presence of the jury:)

THE COURT: The press has asked to see the exhibits, and I am sending out an order asking for the help of the lawyers on this. So just look at it when it comes out and see if you can't help the press follow some of the testimony and the exhibits.

All right. Any issues that you-all want to take up with me?

MR. VAN NEST: Not this morning, Your Honor.

MR. BICKS: No, Your Honor.

THE COURT: Okay. We were going to, then -- let me hand out to you the rulings on seven or eight of the depo designations.

And we had set this morning for at least the first round of arguments on how we deal with the problem that Oracle is trying to hold Google responsible for willfulness and bad faith during the period that the judgment in this case said that Google was right and Oracle was wrong.

Have you read the brief that was submitted by Google?

MR. BICKS: I read it very quickly, Your Honor.

THE COURT: All right. Are you prepared to argue that now?

MR. BICKS: I would like, if I could, to have a chance to study it closer.

THE COURT: All right. Then we'll -- I'll do this: Over the weekend -- didn't I ask for briefing on this by -- on this issue on Monday as well? It seems like I did.

MR. VAN NEST: I think you did, Your Honor.

MR. BICKS: Right.

THE COURT: I would like for both sides to file anything more that you're going to file by Monday early. What did I say in that memo? What time is it due?

MR. VAN NEST: You said noon.

MR. BICKS: Noon.

MR. VAN NEST: We could file it --

THE COURT: All right. Well, at least by noon and earlier would be good, and try to address all of the other decisions -- my own law clerk has found decisions that Oracle failed to cite which go the other way and which say that -- and you did not cite the *Kmar* decision that Google has cited, so I want to give you the full opportunity to address those.

And don't just cite the ones that help you. Cite the ones that go against you and help me understand if they're distinguishable so that I can have the full advantage of this.

I want to circle back to the main -- there are two problems. One is, the jury is going to figure out -- we've got a lawyer on the jury, for goodness sakes. They're going to

figure out that there was a prior trial, and it's probably better for us to control what they learn about that than for that to be done through speculation. That's one point.

The other point is that the -- it is disturbing and it would be a manifest injustice probably -- that's what I'm worried about, I'm asking you would it be -- to leave the impression with the jury that the law has been clear from day one and there never was any judgment to the contrary, and to ask the kind of questions that Ms. Hurst asked yesterday and the kind of thing you said in your opening statement, to leave the impression that Google has just -- that there never was any contrary judgment or law. That's what you're saying to the jury, and you need to provide me with some better authority than you have so far.

MR. BICKS: Yes. We'll do that, Your Honor.

THE COURT: All right. So I'm going to give you that chance to do that. But I feel this is -- that you have brought this on yourself by overreaching. There were ways to deal with this problem that would have gotten you 90 percent of the way to where you want to be, and you failed to do that. You were greedy and went for that extra 10 percent.

So you've created this problem, and you need to help me find a way to deal with it. It would be wrong to leave a false impression with the jury, and that's what you're asking the Court to do.

MR. BICKS: Well, we'll address it, Your Honor. We're not -- I don't believe we're leaving a false impression, and I think the evidence that's been presented, particularly yesterday, is that Google and its executives knew all along that there was a problem here, and they knew it going back to 2006, 2007. That's what the evidence shows. That's what the documents showed.

And nobody is going to come in here in good faith and say that they relied on that period of time between Your Honor's overturning the jury verdict and then the Federal Circuit reversing here. There's not going to be a witness who in good faith is going to come in here and say during that little window of time --

THE COURT: It was two years.

MR. BICKS: -- two years that they then made a decision not to do something or to do it, because --

THE COURT: Well, I expect that they would. I expect -- I don't know. I was going to ask. Do you have a witness who would say they relied on the -- on that --

MR. VAN NEST: Sure. Sure. But we've told our witnesses they can't talk about the prior trial or that reversal, or any of that, based on the agreement that we made. So --

MR. BICKS: So then, Your Honor, what's going to happen when the Federal Circuit -- we talk about the date of

the Federal Circuit decision and then you have conduct by Google after the Federal Circuit decision?

THE COURT: Correct. See, I think that would be -- I think the way -- if I'm reading this *Kmar* decision correctly, which I haven't seen your brief on it yet, but the way it would come down, arguably, would be to say to the jury, Oracle is seeking to -- you know, they've quoted you where you -- is seeking right up until the present and at all times to say that Google was in bad faith, but there was a period of time where the law was -- the law became clear in this case in 2014.

MR. BICKS: Yes.

THE COURT: No question about that. The Federal Circuit made that very clear; but prior to that, the law was exactly the opposite in this case for two years; and then prior to that -- you know, I'm not going to tell the jury this part, but there really was no law on this point. It was just, as one of the witnesses said, folklore because no one had ever brought this kind of case before.

MR. BICKS: Right.

THE COURT: So here we were -- they were -- anyway, so we would have an instruction along the lines of what the Ninth Circuit approved in the *Kmar* case.

MR. BICKS: Yeah. Well --

THE COURT: So you should look at the *Kmar* case and tell me why that -- maybe it doesn't apply. I don't know. But

I expect that these witnesses from Google would say of course they relied for that two-year period on the judgment of the court.

MR. BICKS: Well, if they did that, Your Honor, it would contradict the discovery answers they provided in this case when we asked them what they relied on, and nobody mentioned anything about the decision. So I'll provide that to the Court.

THE COURT: Well, that would be most interesting. Is that true, Mr. Van Nest?

MR. VAN NEST: I'm not sure what counsel is referring to, but I can tell Your Honor we have advised our witnesses, consistent with your orders, not to discuss the history, not to discuss the appeal, not to discuss OpenJDK, which you ruled was out as well. So there would be a lot to talk about --

THE COURT: I didn't rule OpenJDK out for all purposes. I just said that project in 2015 was out.

MR. VAN NEST: Right. Right. That's what I'm referring to. That's what I'm referring to.

THE COURT: So I thought that was a low-blow by you to do that at the last minute and not give Oracle a fair shot a it. So, you know, but I didn't rule OpenJDK out earlier than that.

MR. VAN NEST: No, but you ruled that out, and that's what I'd be referring to.

THE COURT: Okay. Well, all right.

I had another question. Oh, here's another question. Not to make your lives any more complicated, and I don't know the answer to this so don't say that I'm saying it would waive the attorney-client privilege. But if somebody from Google said, "Oh, yeah, we relied on the judge's opinion and the judgment of the Court for two years," then two questions would come up. One would be: Okay. What did the attorneys tell you? Maybe. Maybe that would -- I don't know. Does that waive -- does that open the door to attorney-client opinions?

MR. VAN NEST: I --

THE COURT: Wait. Wait. Don't.

And then the second one is: How would you answer -- well, then how come you didn't switch whenever the -- well, maybe you did. Maybe that's what the OpenJDK was.

MR. VAN NEST: Bingo. Bingo.

THE COURT: Well --

MR. VAN NEST: You figured it out. That's exactly right.

THE COURT: All right. Well --

MR. VAN NEST: And we've told our witnesses they --

THE COURT: That's a lot of time between 2014, and it took you a long time to get that thing done.

MR. VAN NEST: Well --

THE COURT: I wish you had done it three months

earlier.

MR. VAN NEST: -- there's testimony about that in the record as well.

But my point is that we've carefully instructed our witnesses that when they're asked has anything changed, they can't say anything about OpenJDK at this point and they can't say anything about the Court of Appeal decision.

THE COURT: Well, if the door gets opened. If Oracle were to ask that question --

MR. VAN NEST: I think it's been opened, but --

THE COURT: -- then, of course, they can answer it, but you should not put it forward.

MR. VAN NEST: I think it's been opened now, Your Honor, but we haven't made a decision on that yet.

THE COURT: All right. Mr. Baber, you've been hovering around like you're dying to say something. What is your heartburn this morning?

MR. BABER: Nothing, Your Honor. Just in case you had any questions.

MR. VAN NEST: He's going to discuss the instruction, Your Honor, but I understand we're going to pass that until --

THE COURT: Let's pass it, then, until everybody is ready.

But I raised some questions that I'm thinking about, and there would be ways even now for Oracle to agree to scale back

its arguments and to cure this problem, but they haven't proposed it. And if you put me in a position where I've got to give an instruction, I'm going to do it. So I'm not there yet. I'm going to see how things develop for a while.

MR. BICKS: Yes. Can I ask, Your Honor, that as much as I love that filing cabinet, that we remove it from the sight of the jury?

MR. VAN NEST: He asked me about it, and I thought I had. I twisted it. If he wants --

THE COURT: I think we -- you know, it's a fair point. It's ever-present argument by Mr. Van Nest and his famous file cabinet. You know, it would be like if you had the Bible sitting on your desk ever present and visible to the jury. I think we should take it out. You don't have to do it today, but next week let's remove it from the courtroom --

MR. VAN NEST: Thank you, Your Honor.

THE COURT: -- until you want to bring it back for closing argument or with a witness. That would be okay.

MR. VAN NEST: Thank you.

THE COURT: I do want to say one thing. The Oracle brief criticized me for saying to Mr. Schwartz "welcome again." Listen, you can't tell me how to run the courtroom. I welcome every witness. I try to anyway. I try to extend common courtesies to witnesses. And in this case, I did say "welcome again." It was not meant to communicate to the jury that there

had been a prior trial. It was just a common courtesy.

And it just illustrates how artificial what Oracle is trying to do here to conceal from the jury that there was a prior trial and these witnesses have been here before. I am trying to honor what you are asking me to do, but I didn't do that on purpose. It was just the way I would normally greet somebody.

And for you to put that in your brief as if I am somehow sabotaging and prejudiced against Oracle is just ridiculous. The number of things I have ruled in your favor on and against Google is a very long list, and it's just not right for you to pick up on some picayune thing like that and criticize the judge for that. So think about it, would you?

This is -- it's hard to run a trial like this with lawyers and especially when the lawyers don't quote all the authority and they slant the authority in ways that it's like the fog of war. The fog of war. And it's very hard for me to see what even what the law is when I can't get the lawyers to tell me. You know, like you leave the *Kmar* decision out of your brief. I had asked you for the right authority.

Both sides have got to help the judge understand what the law is. Both sides can't try to tell me how to greet a witness.

All right. You don't have to respond unless you want to.

MR. BICKS: I can say, Your Honor, that no one was

taking any potshots. The point that the parties agreed we weren't going to say anything about the prior proceedings.

THE COURT: That's you, not me. That's you. That was between you, not the way the judge runs a courtroom.

MR. BICKS: Understood.

THE COURT: So far I've tried to avoid any reference to that being made out of deference to the lawyers who said that they would prefer for that not to happen; but if I inadvertently say "welcome again" to a witness who has been here before, well, too bad. I'm doing my best.

MR. BICKS: Understood. Thank you.

THE COURT: All right. Are we ready now to continue on with the -- bring in the jury? They may even all be here.

MR. VAN NEST: I believe we are, Your Honor.

THE COURT: All right. Hand up to me the thing you wanted me to read before we get to the -- see if the jury is all here, please.

MR. VAN NEST: We're going to play some video first from Mr. Ellison and Mr. Duimovich, but we were hoping Your Honor could read this before Mr. Bloch. He would be the first live witness.

THE COURT: All right. And you've shown this -- have you shown this to counsel?

MR. VAN NEST: I haven't.

THE COURT: I think I actually have this.

MR. VAN NEST: It's the statement that we worked out.

THE COURT: All right. So it would be starting with agreed statement.

MR. VAN NEST: Right.

THE COURT: Okay. But this would be before Bloch?

MR. VAN NEST: Right.

THE COURT: All right.

MR. VAN NEST: He will be our first live witness, Your Honor, after the -- we're going to play two short videos and one reading of --

THE COURT: Do you have any live witnesses today?

MR. VAN NEST: We've got a lot. We're hoping to get through three or four.

THE COURT: Progress. Okay. You've only gotten through three; right?

MR. VAN NEST: We have, but we're moving well. We have a mix of live and video today.

THE COURT: Okay. Are you-all doing what I asked and giving the clerk the CD of the testimony?

MR. VAN NEST: Our plan was to wait until it was played, and then we will provide a CD --

THE COURT: All right. That's okay, but you've got to do it after --

MR. VAN NEST: -- but, yes.

THE COURT: -- and it has to have just what was

played. You can't have extraneous things on there.

MR. VAN NEST: That's why we're waiting.

THE COURT: Okay.

MR. VAN NEST: That's why we're waiting. We want to make sure we put on the CD exactly what was played and what it is.

THE COURT: Great.

MR. BICKS: We've done that, Your Honor.

MR. VAN NEST: One CD per witness.

MR. BICKS: We've prepared the CDs and given it to them to verify it.

THE COURT: All right. Good.

(Proceedings were heard in the presence of the jury:)

THE COURT: We're about to jump in to some more testimony and witnesses. Thank you for being so punctual. This is before you're supposed to even be here. You're getting earlier and earlier. Thank you for that.

You know, I look over at the jury box and I see how hard and how closely you're paying attention, and I want to thank you for that. And I usually say to juries in civil cases -- I haven't yet in this case, but I just omitted it by accident -- if you want to ask a question, you can write it out, and then give it to the clerk and the clerk will give it to me and then I'll share it with the lawyers so the lawyers will know that something -- I'll give you an example.

You might say, "Was so and so at the meeting on" -- I'll make up a date -- "April 1, 2007?" I'm making that up. And you might say, "Was so-and-so at that meeting?" Because it might make a difference to you and you'd like to know or you're just curious. Just write it out and give it to the clerk.

Now, sometimes if you ask a question, we can't answer it right away and we may not ever answer it. It may be impossible. In my experience, about half the time to three fourths of the time, something like that, the lawyers figure out a way to answer the question.

And it won't be that they turn to you and say, "Thank you for your question. Here's the answer." No. It will just come out in due course through a witness or a document.

And it may be that you don't get the answer for a couple of days, or you may not really -- you might get the answer and you might not realize you've got it.

So we don't allow the lawyers to send a message back in to you in the jury room saying, "Here's the answer to your question." No. Eventually it may come out in testimony is all I'm saying.

One of the reasons we've got this timeline and we give you the notepad is to try to help you understand the evidence in the case. And, you know, a case with technology, it's not always easy. So this is a way for you to tell us a question that you would like to see answered in the case.

But you don't have to send any notes. I would say most jurors do not send out notes, but it's done often enough that I would be happy to consider it with the lawyers and see what we can do.

I'll just leave it at that. And I want to make clear that the option is there, but I'm not asking you to do it. I'm just saying that if you want to do it, it is done in other cases. You can do it in this case, too.

So you will remember that yesterday Mr. Rubin was on the stand for quite a while, and now he's gone, and we're going to go to the next witness.

So, Mr. Van Nest, your next witness, please.

MR. VAN NEST: Good morning, Your Honor. Thank you. And good morning, everyone.

At this time we'd like to present the sworn prior testimony of Mr. Larry Ellison. And, Your Honor, some of the testimony was videotaped and we'll play that, but some of it was not and I'd like to read that. My partner, Mr. Ragland, will provide the answers and I'll provide the questions just as they appear in the transcript.

THE COURT: All right. Okay. So you're Mr. Ragland?

MR. RAGLAND: Yes.

THE COURT: All right. You can come up here and sit down.

MR. RAGLAND: Thank you, Your Honor.

THE COURT: Welcome to the chair.

Okay. Now, here's the deal. We're going to have what's called a reader. He is not under oath. He is just a reader. He's going to read it exactly. He doesn't have to be under oath because the other side has the same thing that's being read from, and they can make sure he's doing it right.

We use a reader sometimes in order to help keep clear who is asking the question and what the answer is. It just brings it more to life for the jury. But the testimony -- so the inflections don't matter or anything. It's just the straight testimony, and they're not supposed to do any inflection, any drama, none of that. It's just a straight Q and A, question and answer, question and answer.

So that's why he's not under oath, but he is a lawyer and I told you nothing that a lawyer ever says in the courtroom is evidence. This is an exception because he's reading exactly what the testimony was and that testimony that comes from the witness stand is testimony. It will be evidence in the case.

Any questions about this procedure? If so, raise your hand and ask.

All right. This will probably happen several times in the trial, and this is the first time so I want to explain it.

All right. Tell us again the date of the prior testimony.

MR. VAN NEST: The date of this transcript is April 17th, 2012, and we'll have a video shortly after that

that was taken earlier than that.

THE COURT: All right. Okay. So you may proceed.

MR. VAN NEST: May I give just a brief preview of what we're going to be discussing, Your Honor? Very short.

THE COURT: Well, is it -- all right. Go ahead.

MR. VAN NEST: All I was going to say is, obviously, this is Mr. Ellison's testimony. He was the chief executive officer at Oracle during the period in question, and he's going to be discussing in the testimony and on the videotape Oracle's efforts to build a smartphone, Sun's efforts to build a smartphone, and something about Java revenues at Oracle after Android came out. Those will be the subjects that he will be covering between the tape and the video -- between the reading and the video.

THE COURT: All right.

MR. VAN NEST: Thank you, Your Honor.

THE COURT: Now, I need to remind you that nothing Mr. Van Nest just said is evidence. It's in the nature of an opening statement. I guess I can allow that, what happened, and I'll let -- Mr. Bicks, you can have the same kind of privilege when we come to any read-ins that you have.

All right. Go ahead, Mr. Van Nest.

LAWRENCE J. ELLISON,

called as a witness for the Defendant, having been duly sworn, was examined and testified through deposition testimony as

follows:

(Transcript of deposition testimony as read by Robert Van Nest and Steven Ragland:)

"Q. Good morning, Mr. Ellison.

So do you know, in fact, Mr. Ellison that Android was released originally back in 2007? Is that something you know?

"A. Approximately. I didn't know if it was 2007 or early 2008.

"Q. And in 2008, the first Android phone came out?

"A. That's what -- that's what was in my head was 2008, the first Android phone.

"Q. And then you entered into an agreement. It's not on this timeline, but you entered into an agreement to acquire Sun in about April of 2009; correct?

"A. Correct.

"Q. And then -- and you then met with Mr. -- the deal closed sometime in early 2010?

"A. Yes.

"Q. Now, by that point in time, you had already appeared at JavaOne and said you were flattered by Android's use of Java.

"A. Yeah. I said I was very excited that there was a Java phone on the marketplace, yes.

"Q. And you said that you expected more Java-based

products from your friends at Google; right?

"A. I don't recall saying that, but --

"Q. Do you recall making the statement that you were flattered by Android?

"A. That I was flattered that there was a Java phone on the market, yes.

"Q. And that Java phone you're talking about was Android?

"A. Yes.

"Q. Right. You said it was exciting?

"A. Yes.

"Q. It was a Java phone; right?

"A. Yes. Yes.

"Q. And you expected to see more Java devices coming from your friends at Google; correct?

"A. I mean, I'll take -- I don't recall that part, but okay.

"Q. Do you remember actually going up on stage with Mr. McNealy prior to this meeting with Mr. Schmidt and making these comments in public?

"A. I do.

"Q. Do you remember your remarks that day?

"A. Yes.

"Q. What I'd like to do is play an excerpt of your remarks and ask you to identify them and authenticate them."

MR. VAN NEST: At this time, we would play TX2939.1.

THE COURT: Is that admitted in evidence?

MR. VAN NEST: It will be.

(Whereupon, the video was played for the jury.)

(Transcript of deposition testimony as read by Robert Van Nest and Steven Ragland continued:)

"Q. Mr. Ellison, is that you on the video?

"A. Yes, it is.

"Q. Now, were your remarks that day recorded and filed with the Government?

"A. I have no idea.

"Q. All right. Do you see the remarks there that are attributed to you in the middle of the page on Exhibit 2041?

"A. I do.

"Q. Do those appear to be the remarks we just heard about on the video?

"A. Can I have a second to look?

"Q. Sure.

"A. They are.

"Q. Is it often the case that sometimes when you make public statements, because of the securities rules, those statements have to be filed with the Government?

"A. Yes.

"Q. That happens from time to time?

"**A.** It does.
"**Q.** Does this appear to be one of those occasions?
"**A.** It is.
"**Q.** And there in the middle of the page, are those the remarks that you made at JavaOne in 2008 -- excuse me -- 2009?
"**A.** They are.
"**Q.** And you said: 'James, Sun has done a fantastic job opening Java, opening up Java, giving Java to the world, and we're going to do more of the same.' Do you see those remarks?
"**A.** Yes.
"**Q.** Those are remarks you made that day?
"**A.** Yes.
"**Q.** Was this a reference to being flattered by the fact that Android was using Java?
"**A.** Yes.
"**Q.** You're talking about an Android phone being shaken. You meant out in the audience?
"**A.** Correct.
"**Q.** And you indicated you were very excited by Android?
"**A.** Yes.
"**Q.** And you liked the fact that it used Java?
"**A.** Yes.
"**Q.** And that you expected Google to do more?

"**A.** Yes.

"**Q.** And to expand its use of Java in Android into netbooks and other products; right?

"A. Yes.

"**Q.** And that was made at a developer conference with thousands of people in the audience?

"**A.** Yes.

"**Q.** The press were there?

"**A.** Yes.

"**Q.** Right. Other companies were there?

"**A.** Yes definitely.

"**Q.** Java developers were there?

"**A.** Lots of them."

MR. VAN NEST: Your Honor, that concludes our reading, and now we will play the video portion of the testimony.

THE COURT: All right. Thank you, Mr. Ragland.

MR. RAGLAND: Thank you, Your Honor.

(Whereupon, the video deposition of Lawrence J. Ellison was played for the jury not reported.)

MR. VAN NEST: Your Honor, that concludes the video portion.

I would like to move into evidence the exhibits that were shown during Mr. Ellison's exam.

THE COURT: All right.

MR. VAN NEST: They are TX2041, TX2042, TX2043,

TX2044, and TX2939.1. That's the video.

THE COURT: Say the last one again.

MR. VAN NEST: 2939.1.

THE COURT: Any objection to those?

MR. BICKS: No, Your Honor.

THE COURT: All received. Thank you.

(Trial Exhibits 2041, 2042, 2043, 2044, and 2939.1 received in evidence)

MR. VAN NEST: We have a second short presentation of video testimony, and Mr. Bayley is going to introduce that.

THE COURT: All right. While you're coming forward, you're just going to read it?

MR. BAYLEY: Oh, no. This is a video we're going to play.

THE COURT: All right. On all these play videos and the designations, after the jury is in for the break, you need to give me the breakdown between who designated what so I can charge the time appropriately. All right?

MR. VAN NEST: We will do that. We have that available, Your Honor.

THE COURT: All right. Thank you.

What is your name again?

MR. BAYLEY: Ed Bayley.

THE COURT: All right. Mr. Bayley, what are you about to do?

MR. BAYLEY: I'm about to play a video of Mr. John Duimovich, which I'd like to introduce to the jury, if that's okay with Your Honor.

THE COURT: What do you mean "introduce to the jury"?

MR. BAYLEY: As Mr. Van Nest described, just describe Mr. Duimovich, who he is, and what he's expected to testify about in the video.

MR. BICKS: Your Honor, I don't have a strong objection to it, but it's supposed to be the way we would normally do with a witness.

THE COURT: You would normally designate the testimony. Just do this: Say what his name is, the date of the deposition, and where the guy works, and who took the deposition. That's really all you should do.

MR. BAYLEY: All right. That's fine, Your Honor.

Good morning. My name is Ed Bayley. I'm one of the attorneys for Google.

As our next witness, we're going to play for you a video of prior testimony from Mr. John Duimovich. Mr. Duimovich is a distinguished engineer and Java chief technology officer --

THE COURT: Well, no. Now you're giving a speech.

MR. BAYLEY: Sorry.

-- with the IBM corporation.

THE COURT: No, no. Just say he works at IBM -- all right -- or did work at IBM.

MR. BAYLEY: He continues to work at IBM, and --

THE COURT: Now you're testifying. At the time of the deposition --

MR. BAYLEY: At the time of the deposition, which took place in December of 2015, he worked at IBM.

THE COURT: All right. Spell the witness' name slowly for the jury.

MR. BAYLEY: Sure. John Duimovich, D-U-I-M-O-V-I-C-H.

THE COURT: All right. Okay. So, in essence -- and what was the date of the depo?

MR. BAYLEY: It was December 2015. I apologize, Your Honor. I do not remember the exact date.

THE COURT: All right. So it was a few months back. And this is somebody who worked -- at the time of the deposition, still worked at IBM; is that right?

MR. BAYLEY: That is correct.

THE COURT: All right. So we'll now roll the tape.

(Whereupon, the video deposition of John Duimovich was played for the jury not reported.)

THE COURT: Is that it?

MR. BAYLEY: Yes, Your Honor. That concludes the presentation.

I'd like to now move into evidence the exhibits that were used with the witness. This is trial Exhibit 7346.

THE COURT: Before we do that, Ms. Hurst, did you have

an issue to raise before we do that?

MS. HURST: Your Honor, perhaps we should make it clear for the jury that the Mr. Ellison referred to in that testimony was not Mr. Larry Ellison of Oracle but, rather, was Mr. Tim Ellison of IBM.

THE COURT: Is that correct?

MR. BAYLEY: That is correct.

THE COURT: That's a very good clarification. The Ellison referred to in that clip we just saw was not Larry Ellison, it was Tim Ellison who works for who?

MR. BAYLEY: IBM.

THE COURT: Just curious. Any relation or totally different people?

MS. HURST: Not that I know.

MR. BAYLEY: Not that I'm aware of.

THE COURT: It happens to be a good point to clear up. Okay. What documents do you want to move in?

MR. BAYLEY: Exhibit 7326, Your Honor.

THE COURT: Thirty-seven -- that's it?

MR. BAYLEY: 7326.

THE COURT: 7326.

MR. BAYLEY: Yes.

THE COURT: Any objection?

MS. HURST: No, Your Honor.

THE COURT: Received in evidence. Thank you.

(Trial Exhibit 7326 received in evidence)

THE COURT: Next witness.

MR. KAMBER: Good morning, Your Honor. Google would like to call Mr. Joshua Bloch.

THE COURT: By depo or screen?

MR. KAMBER: Live. He is here.

THE COURT: Do you want me to read it while he's here?

MR. VAN NEST: You can read it now, Your Honor. He doesn't have to be here.

THE COURT: The lawyers have an agreed-on statement that is like a stipulation. It's not very long. It will take two minutes to read it. This will be evidence in the case, so I'm going to read it slowly, and maybe raise your hand if you miss any of this because this will be the only time you will hear this. It may be read again.

Are you ready over there? You don't have to copy it down word for word, but it's going to relate to what the next witness is going to say. And both sides agreed to this, so this is evidence in the case.

"The Java platform is a software application platform that is used to write and to run programs in the Java programming language. The Java programming language is free and available to use. The Java platform includes, among other things, the Java virtual machine and the Java API packages. 'API' stands for application programming interface."

Counsel, I'd like to read that paragraph again. Any objection?

MR. VAN NEST: No, Your Honor.

MR. KAMBER: No, Your Honor.

MS. HURST: No, Your Honor.

THE COURT: All right. I'm going to do it faster this time, but here we go.

"The Java platform is a software application platform that is used to write and to run programs in the Java programming language. The Java programming language is free and available to use. The Java platform includes, among other things, the Java virtual machine and the Java API packages. 'API' stands for application programming interface."

Now, I'm going to continue on with their agreed-on statement.

"What is at issue in this case are the Java API packages, which are sets of pre-written computer programs used to perform common computer functions without a programmer needing to write code from scratch. These pre-written computer programs assist developers in writing applications. These pre-written programs are organized into packages, classes, and methods.

"An API package is a collection of classes. Each class contains methods and other elements. The packages, classes, and methods are defined by declaring code.

"The declaring code is the line or lines of source code

that introduce, name, and specify the package class or method. The declaring code allows programmers to understand and make use of the pre-written programs in the API packages to write their own programs.

"The declaring code for the packages, classes, and methods reflects the structure, sequence, and organization, sometimes called SSO for short, for the Java API packages."

So let me read that sentence again.

"The declaring code for the packages, classes, and methods reflects the structure, sequence, and organization for the Java API packages. The SSO specifies the relationships between and among the elements of the Java API packages and also organizes the classes, methods, and other elements within the package.

"Each individual method performs a specific function. The declaring code for a method is sometimes referred to as the method declaration or", quote, "'header,'" close quote, "or," quote, "'signature'," close quote.

"The declaring code for a method tells the programmer the information the method needs, the inputs, to perform the desired function. Each method also contains implementing code. The implementing code provides step-by-step instructions that tell the computers how to perform the functions specified by the declaring code.

"The declaring code and the SSO of the 37 Java API packages at issue are protected by copyrights owned by Oracle.

The copyright protection does not extend to the idea of organizing functions into packages, classes, and methods, but the copyright protection does cover the SSO as expressed in the 37 Java API packages."

So before we go any further, if you want me to read that again, or parts of it, raise your hand.

Okay. Four hands went up.

I'm going to go back to the part that I -- I'm not going to repeat what I did read twice, but I'll go back, and this time I'll read a little faster, but I will read again what I have already read, except for the part that I have already read twice.

"What is at issue in this case are the Java API packages which are sets of pre-written computer programs used to perform common computer functions without a programmer needing to write code from scratch. These pre-written computer programs assist developers in writing applications. These pre-written programs are organized into packages, classes, and methods.

"An API package is a collection of classes. Each class contains methods and other elements. The packages, classes, and methods are defined by declaring code. The declaring code is the line or lines of source code that introduce name and specify the package, class, or method.

"The declaring code allows programmers to understand and make use of the pre-written programs and the API packages to

write their own programs.

"The declaring code for the packages, classes, and methods reflects the structure, sequence, and organization for the Java API packages. The SSO specifies the relationship between and among the elements of the Java API packages and also organizes the classes, methods, and other elements in the package.

"Each individual method performs a specific function. The declaring code for a method is sometimes referred to as the method declaration, header, or signature. The declaring code for a method tells the programmer the information the method needs, the inputs, to perform the desired function.

"Each method also contains implementing code. The implementing code provides step-by-step instructions that tell the computer how to perform the functions specified by the declaring code.

"The declaring code and SSO of the 37 API packages at issue are protected by copyrights owned by Oracle. The copyright protection does not extend to the idea of organizing functions into packages, classes, and methods, but the copyright protection does cover the SSO as expressed in the 37 Java API packages."

All right. Maybe you will hear that again at some future point, but I think that's the best I could do for now.

Please call your next witness.

MR. KAMBER: Thank you, Your Honor.

Google calls Mr. Joshua Bloch. He was just outside the door a minute ago.

THE CLERK: Will the witness please approach the witness stand.

THE COURT: All right. Are you Mr. Bloch?

THE WITNESS: I am.

THE COURT: Okay. Welcome. Please come up here and raise your right hand and take an oath to tell the truth.

JOSHUA BLOCH, DEFENDANT WITNESS, SWORN

THE CLERK: Please state your name for the Court, and spell your last name for the record.

THE WITNESS: Joshua Bloch, B-L-O-C-H.

THE COURT: All right. Great. Welcome. Please have a seat.

And you should adjust the mic so it's about this close to your voice. That's right. Say your name.

THE WITNESS: Joshua Bloch.

THE COURT: That's very good.

Counsel, go ahead.

MR. KAMBER: Thank you.

Again, ladies and gentlemen, my name is Matthias Kamber. I'm one of the lawyers for Google. Mr. Van Nest introduced me earlier this week, but it's been until today where I get --

THE COURT: Spell your name for the jury.

MR. KAMBER: Sure. It's M-A-T-T-H-I-A-S, H is silent.

K-A-M-B-E-R.

THE COURT: Kamber?

MR. KAMBER: Kamber.

THE COURT: All right. Mr. Kamber, go ahead.

MR. KAMBER: Thank you.

DIRECT EXAMINATION

BY MR. KAMBER:

Q. Good morning. Welcome.

A. Thank you.

Q. Again, can you please introduce yourself to the jury?

A. Yeah. My name is Josh Bloch.

Q. Can you please tell the jury a little bit about your educational background?

A. Sure. I went to Columbia University in New York. I graduated in 1982 with a B.S. in computer science. And then I went to a school called Carnegie Mellon University in Pittsburgh, Pennsylvania, and I was there until 1990 when I graduated with a Ph.D., also in computer science.

THE COURT: Mr. Bloch, when you look at the jury, your voice goes away from the microphone. If you're going to do that, that's fine, but just pull the microphone over so everybody, including the public, can hear you.

THE WITNESS: Excuse me. I'll do that.

THE COURT: Thank you.

BY MR. KAMBER:

Q. Dr. Bloch, what do you do now?

A. I'm a professor of computer science at Carnegie Mellon University.

Q. What classes do you teach?

A. I teach a class called principles of software construction.

Q. In terms of your career, what did you do after you finished your Ph.D.?

A. First I did a start-up company with my thesis adviser in Pittsburgh, and that lasted about six years. And then I moved on to JavaSoft, Sun Microsystems.

Q. And when you joined Sun in 1996, did you know the Java programming language?

A. Actually, I didn't.

Q. How did you learn the Java programming language?

A. Well, I -- I learned it just about when it first came out. I got one of the first books on the language, which was called *Java in a Nutshell*, and I read that book on the airplane to California on my way to the job interview.

Q. When you used that *Java in a Nutshell* book, what, if any, discussion of APIs did that book have?

A. It discussed them extensively. It actually contained the documentation of all of the APIs in the language at that time. I think Flanagan, who wrote the book, described that section as

the heart of the book.

Q. How easy or hard was it for you to learn the Java language?

A. It was actually pretty easy, and the reason is that the Java language and its libraries made extensive use of ideas and APIs from previous languages. So, for example, the entire math API in Java at that time was the same as the math API from the C programming language that I already knew. So when I got to that chapter, I said, "This is great. I don't even have to read this chapter because I already know this API," and I just moved on.

Q. Can you give an example of an API that was in that math class in Java that was also in C?

A. Sure. There was, you know, pow was the name for the exponentiation function, and it isn't even a good name, but they kept that name because programmers all knew it.

Q. When you were at Sun, Dr. Bloch, what was your job title?

A. By the time I left, I had been promoted to distinguished engineer, and I was also the Java platform library's architect.

Q. And what were your duties generally at Sun?

A. I designed APIs. I implemented them as libraries. I wrote documentation for them. I tested them. I actually went around the country giving lectures, sometimes at Sun-sponsored conferences like JavaOne at San Francisco. Once even an Oracle-sponsored conference called Oracle Open World also at

San Francisco. I helped junior engineers learn to code and design better. That kind of thing.

Q. You mentioned libraries just now. Can you provide some context? Can you explain for the jury what a library is that you worked on at Sun at your time there?

A. Sure. The libraries that I worked on when I was at Sun basically served to amplify the power of the language. The idea is that --

THE COURT: What language are we even talking about?

THE WITNESS: We're talking about the Java programming language.

THE COURT: You haven't said that in a while. I thought maybe it was some other language. You're talking about Java?

THE WITNESS: It was all Java. I'm sorry. I -- my -- my job was entirely Java at Sun. I occasionally used C in the implementation of Java, but my APIs were all Java APIs.

THE COURT: Well, the rest of us don't know all that, so you keep in mind that we're learning as we go.

All right. Please cooperate with your answer.

THE WITNESS: Sure.

Okay. What was the question again?

BY MR. KAMBER:

Q. The question is if you could explain the libraries or the class libraries that you referred to so the jury understands

what we're talking about.

A. Yeah. So when a programmer sits down to write a Java program, there are certain things that they all have to do, certain just basic tasks. So programs have a lot of, let's say, sorting of lists, searching in lists. You have to write things out to the screen of the computer, write things out to the disk to be stored, and it would be at best tedious and at worst impossible for every programmer to write these low level building blocks themselves so the libraries provide this functionality in nice little packages of code that the programmer can call on, and those are the libraries.

Q. How are the libraries that you worked on at Sun organized?

A. A, it's a three-level hierarchy, so you have packages which consist of multiple classes, and each class consists of multiple methods or functions. You can call them one or the other, but they're the same things, and those are the things you actually call on to do stuff like searching and sorting.

Q. Why did you organize the libraries in that way, in that hierarchy?

A. I actually had no choice in the matter. That's mandated by the language. The language spells that out for you. A computer language is completely inflexible. You -- it has a certain set of rules, and you have to obey those rules; and in Java, all libraries are organized in that way. They are packages containing classes containing methods.

Q. During your time at Sun, Dr. Bloch, how much time, if any, did you spend considering the structure, sequence, and organization of the APIs you were developing?

A. None actually. That is not a word that we even use in the software design community. I didn't hear that phrase until the trial started.

Q. Can you give some examples of APIs that you worked on at Sun?

A. Sure. Yes. With pleasure.

So when I first went out there in 1996, I wrote an API called java.math -- not to be confused with java.lang, dot math, which was in the book I read on the plane on the way over -- and java.math allowed you to do what is called multiprecision arithmetic. That is like doing math on big numbers, like thousands of digits long. And that is very important to cryptography. That is the first API I did.

And the API that I did that is best known is what is called the collections API. A collection is like a bunch of things. So these are APIs that let you operate at one time on, like, a list of words or a mapping, a dictionary that goes from, like, name to social security number. These are called collections. And I wrote those APIs. I wrote a bunch of others as well.

Q. Okay.

MR. KAMBER: Permission to approach, Your Honor.

THE COURT: Yes.

BY MR. KAMBER:

Q. Dr. Bloch, I have just handed you what has been marked as Trial Exhibit 623.11. Do you recognize that document?

A. Yes, I do.

Q. What is it?

A. This is actually the source code for a utilities class in the collections library.

Q. Who authored this code?

A. I did.

Q. How do you know that?

A. There's something called an author tag in the code when you write it. You put down your name. So line 44 on page 1, that says at author Josh Bloch, and that's me. I put that in there when I wrote it.

MR. KAMBER: Your Honor, we'd like to offer TX623.11 into evidence.

MS. HURST: No objection.

THE COURT: Thank you. Received in evidence.

(Trial Exhibit 623.11 received in evidence)

BY MR. KAMBER:

Q. We are publishing it for the jury here. You may see it on your screen there, Dr. Bloch, in front of you.

Using this collections class that you wrote, can you please explain to the jury how it was that you put methods that

you wrote into a particular package or into a particular class?

A. Sure thing.

So if you look at the first page on line 8, you'll see it says packagejava.util, and that says that this class is part of the package, which is the top level in that three-level hierarchy called java.util.

Q. What is that called, by the way?

A. That is called a package declaration. You are declaring that this class is part of the package. Okay?

And then the next thing is that on -- I'm going to say immediately -- so what you see from lines 14 all the way down until 53 is just a comment. That's not code at all. That's not a program. It's just sort of information.

And then the next real line of the program is line 54.

I'm sorry if I'm too close to the mic.

But, anyway, line 54 it says public class collections, and that's the line that declares that this class is called collections. It has the name collections, and the public means that it's part of the API that anyone who is using this library can use the methods that are declared in this class.

Q. So what is this line 54 called?

A. It is called the class declaration, just as the one at the top was called the package declaration.

Q. Can you give the jury an example of a method declaration in this collections framework that you wrote?

A. Yeah. I could do that. Give me a moment.

All right. So I'm looking for the reverse method, which would be a good one to show you. It's on page 7. If you go to page 7 and you look at line 366, you'll see there's a method declared there called reverse. The line says "public static void reverse list of question mark list."

Q. So what is that that the jury is looking at?

A. That does sound like gobbledegook, but it's not. "Public" means this method is part of the API, just as the public for the class said that the class was part of the library.

"Static" means that it takes all of its input parameters between those parentheses after the word "reverse" and it only has one input parameter.

The "void" means it doesn't return any value other than changing its input parameters potentially.

"Reverse" is the name of the function, and it's descriptively named as you'll see in a moment.

And then "list of question mark" means you can pass in a list of anything, whether it's a list of words or a list of numbers.

And that -- that's what it means. And what it does is it reverses the order of the elements in a list. So, for example, if you called it with a list that said Romeo loves Juliet, then the function call would look like collections.reverse of my list; and after the call, it would say Juliet loves Romeo.

So it's actually pretty simple what it does. It just reverses the order of the elements in the list, whether they are word or numbers or whatever they happen to be.

Q. How does the method know how to do that, that reversal?

A. Ah. Below the declaration is a lot more code. Let's take a look.

It starts on line 367 and it goes all the way down to 380, and that is the actual code that tells the computer how to do it. It's a little bit complicated. There's the two different ways depending on what kind of a list it is, but the basic idea behind how it works isn't even too hard.

The idea is if you want to reverse a list, one way to do it is to take the first element of the list and the last element and you swap them; and then you take the second element of the list and the second-to-the-last and you swap them; and you keep swapping elements until you get to the middle.

And when you're done with that, you'll find out that that's reversed the order of the elements in the list. So that's basically what we've instructed the computer to do in this sort of complicated-looking implementation code.

Q. So, Dr. Bloch, we're looking here. We have lines 366 through 380 in the collections class; correct?

A. Yep.

Q. And that's the reverse method?

A. Yeah. That is the reverse method.

Q. Okay. So, again, so the jury understands this, the top line is what?

A. The top line is the method declaration.

Q. Okay. And the lines 367 through 380, what are those?

A. Those are the implementation. They're the code that actually tells the computer how to do it.

Q. Okay. If you change the implementation, to what extent, if at all, do you have to change the declaration for the method at the very top?

A. So that's the whole beauty of APIs. You don't have to touch the method declaration. The method declaration basically forms the nexus between the code that calls the library and the code that does the work. So you can change the implementation all you like. You can do an implementation. You know, it can be a completely different technique, but as long as it reverses the list, the client code that is the calling code is none the wiser.

Q. As between the APIs that you wrote -- that is, the method declarations and the implementing code that you wrote -- would you characterize one as more important than another?

A. Not really. You need both of them. It's not going to run unless you have both the declaration and the implementation. They're both. In some cases it's harder to write the APIs, and in some cases it's harder to write the implementations.

Q. When you were writing APIs at Sun, what were you trying to

achieve?

A. I was trying to make the language more powerful, easier to use. I was basically trying to make it easier for programmers who were going to use the language to get their jobs done. You know, I wanted to just make that whole process of writing a Java program that does something useful easier by giving them these new functional tools to use.

Q. What, if any, design principles did you apply when you were writing out method declarations?

A. I've got a whole bunch of them. Important ones include that an API should be as short as possible but no shorter, which basically means you want to provide all the tools that the programmer needs to get the work done but nothing beyond that because anything beyond that is just needless complexity which may confuse the programmer.

You want APIs to be easy to learn and to use but hard to misuse, or maybe even impossible to misuse because you don't want people to have bugs in their programs. So you try to design APIs so that using the APIs won't cause bugs, and there are other principles like that.

Q. To what extent, if at all, would you want somebody to be able to memorize an API; a developer, for example?

A. It's actually very important for a developer to be able to memorize an API. I tried hard.

Q. About how much of your career has been spent on API design

and development?

A. Pretty much all of it. The first API that I actually remember writing was, like, back in the summer of 1983. I worked for IBM Yorktown Heights Research Center as a summer intern, and I wrote an API for a parallel processing IBM 370 that they had there.

Q. For which edition of Java were you developing APIs?

A. It's the one that eventually became known as Java SE for standard edition. Before that it was Java 2 SE, and before that it was just Java, the JDK.

Q. What kind of environment was Java SE being used in during the time you were at Sun?

A. It was being used for desktop computers, servers, you know, powerful laptops, that sort of thing.

THE COURT: Remind us when you were at Sun again.

THE WITNESS: I joined Sun in 1996, and I left Sun in 2004.

THE COURT: Okay. Thank you.

Go ahead.

BY MR. KAMBER:

Q. Dr. Bloch, how, if at all, were the APIs that you developed made public?

A. They are translated into HTML, which you probably know is the basic language of the Web. So when you look at a Web page, you're looking at HTML, and that -- excuse me -- that HTML was

published on the Web, and you can look at it now. Probably not now; but, anyway, it was also -- we -- we -- I wrote books about it and, as I said, I gave lectures about it. So that's how they were published.

Q. Can you explain to the jury why it was that the APIs that you were writing were being made public?

A. Yeah. They were -- they were being made public so people could use them. If you build a tool but you don't tell people about it, you know, might as well not have built it. So you have to both build the tool and you have to tell people, "We've got this new tool for you that you can use."

Q. How were you involved in documenting the APIs that you developed?

A. I documented all of my own APIs. I took great care to do my best to document them well.

Q. Why were -- excuse me.

Why did you want to provide documentation for the APIs you developed?

A. Again, so they could be used. I mean, if you give people a tool but you don't tell them how to use it, you might as well not give it to them.

Q. Were there any other reasons besides having them know how to use the API?

A. Yeah. Sure. Once an API has been documented, then other people can provide their own independent implementations of

that API, and so that's another reason to write good documents. If your documentation isn't good, then people won't be able to re-implement the API.

Q. When you talk about re-implement, again, that was the implementation code that we saw before; correct?

A. Yeah. That -- that was like the code that actually told the computer how to reverse the list. So it would -- like, you could write a new way of doing that.

Q. Okay. To what extent, if at all, did you expect that other people, other programmers, might create independent implementations of the APIs that you developed at Sun?

A. I certainly hoped they would.

Q. Why did you hope so?

A. Because it's pretty much the mark of a successful API. Once an API starts getting reimplemented, you know it has succeeded.

Q. What, if anything, did you do to promote the APIs that you developed at Sun?

A. Well, as I said, I -- I gave lectures. I wrote books. I talked to engineers about them whenever I could. You know, pretty much did everything in my power.

Q. You just mentioned a book, Dr. Bloch. Let me hand you what's been marked as Trial Exhibit 7640. Do you recognize this document?

A. I do.

Q. What is it?

A. This is *Effective Java*, which is a book that I wrote about the Java programming language.

Q. Okay. To what --

MR. KAMBER: Well, let me just move that book into evidence. At this point, Your Honor, we would move TX7640 into evidence.

MS. HURST: It's hearsay, Your Honor, and not admissible under 705.

THE COURT: Is it being offered for the truth?

MR. KAMBER: It's being offered to demonstrate that the APIs were discussed in conjunction with the language, Your Honor.

MS. HURST: Your Honor, this witness is not disclosed as an expert.

MR. KAMBER: That's fact testimony, Your Honor.

THE COURT: Well, what was the year of the book?

THE WITNESS: It was initially published in 2001, and I revised it for a second edition in 2008.

BY MR. KAMBER

Q. Let me ask one other question here, Dr. Bloch. Was the fact that you were publishing this book known to your supervisor at Sun?

A. Yeah. It was pretty much a job requirement.

MS. HURST: May I *voir dire*, Your Honor?

THE COURT: Yes. Go ahead.

VOIR DIRE EXAMINATION

BY MS. HURST:

Q. Is that the second edition, Dr. Bloch?

A. This one is the second edition, yes.

Q. And that's 2008?

A. Yeah.

Q. That was when you were employed at Google, am I right?

A. Yes.

Q. And you haven't mentioned that you were employed by Google for a period of time yet during your testimony, have you?

A. I haven't been asked.

MS. HURST: Your Honor, it's hearsay. He wrote it while he was employed at Google.

THE COURT: Do you have the 2001 version?

THE WITNESS: I have the PDF on my laptop.

MS. HURST: That has not been disclosed, Your Honor.

THE COURT: What's the point you're trying to make with this book?

MR. KAMBER: Your Honor, the point is that the language and the APIs are discussed as being sort of integral with one another, and the book helps demonstrate that. That's all.

THE COURT: Let's pass the book for now, but he can testify to that, but stick to the time period that he was at

Sun.

MR. KAMBER: Sure.

THE COURT: And we'll come back to the book when the jury is not here. So you go ahead, but he can testify to that point verbally.

MR. KAMBER: Thank you, Your Honor.

DIRECT EXAMINATION (resumed)

BY MR. KAMBER

Q. Dr. Bloch, you mentioned writing a book while you were at Sun; correct?

A. Uh-huh.

Q. What was that book called?

A. The book was called -- actually at the time it was called *Effective Java Programming Language Guide* and it was shortened to *Effective Java*.

Q. Did the book that you wrote win any awards?

A. It won a whole bunch of awards, yeah.

THE COURT: This is which one?

THE WITNESS: The first edition.

BY MR. KAMBER:

Q. So the first edition of your book won a number of awards; correct?

A. It did.

Q. And to what extent did the first edition of the book that you wrote while you were a Sun employee also contain discussion

of APIs?

A. It contains extensive discussion of APIs. It, you know, permeates the text.

Q. Instead of publishing about the APIs that you developed at Sun, did you ever consider just keeping them secret?

A. No. That basically would have been contrary to the whole purpose of the APIs. As I said before, the APIs were there to make the language more powerful, to provide new functional tools for users of the language; and if we kept them a secret, nobody would have been able to use them.

Q. To what extent, if at all, did you consider the APIs that you were developing at Sun to be a part of the Java programming language?

A. I considered them to be an integral part of the Java programming language.

MR. KAMBER: Permission to approach, Your Honor.

THE COURT: Yes.

THE WITNESS: Wow.

BY MR. KAMBER:

Q. Dr. Bloch, I've just handed you a copy of a different book. It's been marked as Exhibit 984. Do you recognize that?

A. Yes. I recognize this one, too.

Q. What is that?

A. This is the Java language specifications. So this is the official specification for the Java programming language.

Q. Which edition of the Java language specification do you have before you?

A. This is the third edition.

Q. And for what version of Java does that apply?

A. This applies to Java 5.

Q. And how, if at all, were you involved in Java SE 5?

A. I was involved extensively. I was actually involved more in this edition -- sorry -- this version of the language than any other. I added a bunch of features to the language and a bunch of APIs that were required to support those features.

Q. Was this a publication that was authorized by Sun?

A. Yeah. It says so on the cover. It says "Sun Microsystems from the source." It was most definitely authorized by Sun.

MR. KAMBER: Your Honor, at this point I would move Exhibit 984 into the record.

MS. HURST: No objection.

THE COURT: Received in evidence.

(Trial Exhibit 984 received in evidence)

MR. KAMBER: Thank you.

THE COURT: Just so we'll be clear with the jury -- and I ask both counsel -- is Java SE 5 the same thing as Java SE 5.0 that is one of the copyrighted works in this case?

MR. KAMBER: Yes, it is, Your Honor.

MS. HURST: So stipulated.

THE COURT: All right. So that may help the jury

understand what this document is about.

Okay. Please go ahead.

MR. KAMBER: And maybe I can ask a clarifying question, Your Honor.

Q. With respect to the naming or the -- excuse me -- the numbering of the different versions of Java, how did that work during your time and after when you were at Sun?

A. It's kind of a mess. There is actually a table at the beginning of the first edition of this book which contains a mapping from the internal version numbers that we used at Sun to the public version. So we called it 1.1, 1.2, 1.3; but then outside of Sun, they had marketing names where it was, you know, JDK was the first version and then JDK 1.1; JDK 1.2, also known as Java 2 SE or J2SE; and then, you know, I don't even remember. But I generally just think of them as Versions 1, 2, 3, 4, 5. That's the easiest way to think of them.

Q. Now, we've got the book here. Were you involved in this book?

A. Yeah, I was actually. I -- because I had added these new language features and the APIs that were acquired to support them, I actually gave a lot of prose, that is descriptions of the work that I had done to Gilad Bracha. He is the fourth author listed on the cover. He is the guy that was maintaining the book at that time. The first three authors wrote the first three editions of the book, and then Gilad took over

maintaining many subsequent editions.

Q. You are mentioned in this book correct?

A. I am. I think I'm mentioned somewhere in the acknowledgments.

Q. Let me help you, Dr. Bloch. If you could take a look at page 27, there are some numbers at the very bottom of the document.

A. Got it.

Q. We'll pull that up on the screen. The second full paragraph down, please, Mr. Dahm.

It says here: (reading)

"Another individual who deserves to be singled out is Joshua Bloch. Josh participated in endless language design discussions, chaired several Expert Groups and was a key contributor to the Java platform. It is fair to say that Josh and Neal care more about this book than I do myself."

Do you see that?

A. I do.

Q. You are also thanked and mentioned as the chair of various different efforts and initiatives on the next page; correct?

A. I -- I believe I am. yes. Yes, I am.

Q. So can you just quickly explain to the jury what it was that you contributed to Java SE 5 or 1.5 that's reflected in this book?

A. It's too much stuff to discuss now. I'll give you a few examples.

My two favorite language features that I added were one called the for-each loop, sometimes known as the enhanced for-loop, which made it much easier to iterate over these collections. That is, if you wanted to do something to every single element in a collection, let's say we had that Romeo loves Juliet sentence, if I wanted to capitalize every word, that would be much more easier with this new for-each construct that I added.

I also added something called enum types. When you do have a small group of things, like the months in a year or the nine planets, eight planets in the solar system, I added a facility to the Java programming language that let you manipulate these enumerated types.

So that's a couple examples, and there were plenty others as well.

Q. Let me direct your attention, Dr. Bloch, to there is page 35, again, of the document. There is a Section 1.3. Do you see that on your screen?

A. I do.

Q. It says "Relationship to predefined classes and interfaces." Do you see that?

A. I do.

Q. And the second sentence here says: (reading)

"In particular, some classes have a special relationship with the Java programming language."

Do you see that?

A. Uh-huh.

MS. HURST: Objection, Your Honor. This is getting into that expert testimony we discussed the other morning. It may be that he worked on it at the time, but this is still a matter of specialized knowledge. There was no report. No opportunity --

THE COURT: As long as he sticks to the past tense and what happened back then, overruled, but you cannot start using the present tense because you were not designated as an expert. So as a result, you can explain what you did back then and why you did it. If you worked on this book, if this is a book you worked on, okay. But you've got to stick to the past tense. All right?

THE WITNESS: Very good.

THE COURT: All right. Counsel, stick to the past tense. Objection overruled.

MR. KAMBER: Thank you, Your Honor.

Q. Dr. Bloch, to what extent was it your understanding when you were working on Java SE 5 that there were certain classes that have a special relationship with the Java programming language?

A. I was intimately aware of that fact because I wrote some

of those classes.

Q. And can you explain to the jury what you mean by --

THE COURT: No. What you meant.

MR. KAMBER: What you meant.

Q. What your understanding was -- let me start over, Your Honor.

Can you please explain to the jury what your understanding was with respect to classes having a special relationship to the Java programming language?

MS. HURST: Your Honor, past tense does not solve the 702 specialized knowledge problem. This is going to drive a truck through Rule 26.

THE COURT: I'm sorry. I'm sorry. But this is the rule, and we happen to have a case where the percipient witnesses -- meaning fact witnesses -- the fact witnesses themselves happen to have specialized knowledge and skill. Just like if you had a case that involved doctors and a misdiagnosis or something, all of the percipient people would be very well skilled in medicine, and I can't stop them from talking about their art.

This objection is overruled. You would be correct if he veers off into present-day opinions; but if he's giving opinions on what he actually thought and did back at the time in question, it is fair game. Objection overruled.

Please answer the question.

THE WITNESS: All right. So as I -- do you want to ask the question one more time?

BY MR. KAMBER:

Q. I'll try if I can remember it, Dr. Bloch.

To what extent was it your understanding when you were working on Java SE that there were certain classes and packages that had a special relationship to the Java programming language?

A. Right. Yes. So I -- I understood this to be true. For example, remember before I was talking about that for-each function that I added to the -- not function, but the construct that I added to the language that let you kind of iterate over a whole collection and, you know, like capitalize all the words or whatever.

In order to do this, that collection has to implement a class called java.lang.iterable, and so that class is actually mentioned in this manual here. I can't tell you the page number offhand, but it's in here. I know because I wrote the prose and I gave it to Gilad.

And then in order to do its work, the language has to call a method called iterator on this iterable class, and it returns an object of type iterator.

So that's another class that you absolutely need in order to implement the Java language as specified in this book. You can't do it without that class. And, interestingly, that class

isn't even mentioned in this book. It's what is called an indirect dependency. So the class iterable is mentioned in this book, and you need it to implement this language, and you need the class iterator to implement the class iterable.

So there are actually plenty of classes that are mentioned in this book that you need to implement the Java language. There are about 60 of those, and they bring in a bunch of indirect dependencies. So you need 177 classes at least just to implement the language, and they contain thousands of methods.

Q. Thank you, Dr. Bloch.

I want to show you a document that's been marked as Exhibit 4027. It's another big one. Do you recognize 4027, Dr. Bloch?

A. Yes, I do.

Q. Is this a book that you -- who published this book?

A. Also Addison-Wesley. This is the same Java from the source series. You can see they all look kind of like one another.

Q. What edition of the book is this, 4027?

A. It's the first edition. It doesn't even have an edition number because they didn't know there was going to be a second.

Q. Is this a book that you used during your time at Sun?

A. Yeah. I used it extensively to, you know, write libraries and stuff.

Q. What in particular were you using from in here in order to write libraries, APIs?

A. I -- I was using descriptions both of the syntax and semantics of the language itself and of its core libraries.

Q. Did some of the guidance provided by this first edition specification influence the way you designed any APIs that were in Version 1.4 of Java SE?

A. Yes. It had to. Yeah. Any API that I write has to conform to the rules and the conventions in this book. This is, you know, the Bible, if you will, of the Java platform.

MR. KAMBER: Your Honor, at this point I would like to offer TX4027 into evidence.

MS. HURST: No objection.

THE COURT: Received.

(Trial Exhibit 4027 received in evidence)

BY MR. KAMBER:

Q. Dr. Bloch, I'd like to point you to I think it's page 31 of this document. There's a description in the introduction. It talks about certain of the chapters, and here it's discussing Chapters 20, 21, and 22.

A. Uh-huh.

Q. Do you see that?

A. I do.

Q. Okay. And it talks about it describes the package java.lang. Do you see that?

A. Yes.

Q. And it says that the types defined in java.lang are automatically imported to be available without qualification in all Java programs. Do you see that?

A. I do.

Q. Okay. What was your understanding at the time that you were writing APIs at Sun with respect to the package java.lang being a required package?

A. I -- I think the document speaks for itself here. It's part of the language, and all of its contents are automatically made available to you without even what's called an import statement. That's basically having to ask for them. So most libraries, if you want to use, you have to ask for it. Like, import, you know, I don't know, swing dot whatever. But the java.lang package you get for free because everyone needs it. It's just part of the language.

Q. What about the package java.util that is described -- here is referred to as being described in Chapter 21? How did that package -- what was your understanding as to the relationship of java.util with respect to the Java programming language when you were developing APIs?

A. That's another one which is, you know, very basic. We called them core packages. These were the packages that were most critical to the language.

Q. Was java.io, that's also described here, a core package?

A. Yes. Absolutely. I mean, you know, you can't write a program that actually prints something without it.

Q. And going back to that example that you gave of iterator or iterable, with respect to, just so the jury understands your testimony, with respect to implementing the java.lang, java.util, java.io packages, to what extent, if at all, would you have to use or rely upon other packages and APIs in the Java language?

A. I -- I -- as I say, I -- I believe that you cannot implement those in isolation. I believe you need ten packages just to implement the classes that are described in -- in the third edition of this book.

Q. Now, I want to change gears just a little bit, Dr. Bloch.

A. All right.

Q. How, if at all, have you helped others create independent implementations of the Java APIs that you developed at Sun?

A. When I was working on those collections APIs that I talked about before, I got email from some engineers involved in an open source project called GNU Classpath, and they were trying to re-implement my APIs and they had various requests for clarification. They wanted to know exactly what did you mean when you said this, so I helped them.

Q. Approximately how many times did you communicate with people that were involved in the GNU Classpath project?

A. About five or six. I don't remember exactly.

Q. Was the fact that you were helping GNU Classpath do an independent implementation of the APIs you developed at Sun known to your supervisors?

A. Yeah. I reported to a guy named David Bowen at the time, a really nice guy, and I told him. So he knew.

Q. Did Mr. Bowen approve of the work?

A. Yeah, he definitely approved.

Q. What was your understanding as to why he would approve the work that you were doing with respect to this other competing implementation?

MS. HURST: Objection. Leading, Your Honor.

THE COURT: Sustained.

BY MR. KAMBER:

Q. What, if any, understanding -- let me ask it differently, Dr. Bloch.

What, if any, benefit did Sun get from the Classpath project?

A. As I mentioned before, the people from Classpath would ask me -- when my specs weren't good enough, they would say, "What did you mean here? What exactly should this function do? You know, let's say, you have a function to calculate the minimum element of a list, what should we do if there are no elements on the list?"

And I would say, "Gosh. You know, I'm sorry. I forgot to specify that." And I would add it to the specification, and it

would improve the specification.

As I mentioned, it takes a very, very good specification to admit an independent re-implementation. So by having these guys doing a re-implementation so early, they improved the quality of the specification, and that improvement was usable by every programmer who used this spec to do their programs.

And the other benefit is that when you have multiple implementations of an API, the skill sets transfer so someone who's learned it from GNU Classpath can then transfer those skills to Sun's JDK and vice versa. So it makes the skill set of learning these APIs more valuable if there are more implementations of it out there.

Q. Okay. To what extent, if at all, were you aware of anyone at Sun suggesting that it wasn't acceptable for GNU to do an independent implementation of those declarations that we saw of APIs that you wrote?

A. I was unaware of anyone saying anything of the sort.

Q. When you were at Sun, to what extent were you ever involved in re-implementing an API sort of from the outside world?

A. That same release, Java 5, under a project called nio, we added what was called regular expression processing to the language at the time. And that's a complicated word, but it's just a kind of text processing. When you're dealing with text, it makes things much easier to do.

And it was a junior engineer by the name of Michael McCloskey who actually did the work. And instead of designing our own API from scratch, we decided we would use the regular expression API from this language called Perl 5. It was a -- it was a large, complex, and well-known API to do regular expression handling.

Q. How did you go about doing the re-implementation of that specification?

A. We downloaded the specification from the Web, from their website, and then the engineer, Michael McCloskey, studied it until he understood it well enough to write a new implementation from the ground up without using any existing code.

Q. Why did you choose to re-implement the regular expression API from Perl 5 instead of creating your own?

A. Because it was really quite widely known. It was called a *de facto* standard. Every programmer -- not every programmer, but most programmers who wanted to use regular expressions wanted to use Perl regular expressions, so we transferred their skill set from the Perl language to the Java language by implementing the same API.

Q. To what extent, if at all, did you seek permission from the folks, Perl 5, before doing your re-implementation?

A. To the best of my knowledge, we didn't seek permission at all.

MS. HURST: Objection. Lacks foundation. Calls for a legal conclusion.

THE COURT: When you say "to the best of your knowledge," it begs the question how good your knowledge is. So tell us what you base your statement on that no permission was sought.

THE WITNESS: I didn't seek it. I was -- you know, David Bowen, our manager, didn't seek it, but I don't know that anyone did. It seems unlikely that anyone did. Just based on the way we did things at the time, it seems very unlikely.

THE COURT: If someone had tried to obtain permission, to what extent would you have known about that effort?

THE WITNESS: I almost certainly would have known because we made extensive use of mailing lists, and it would have been on the mailing list.

THE COURT: All right. The objection is overruled. There is sufficient foundation for the testimony.

Next question.

BY MR. KAMBER:

Q. To what extent, if at all, were you aware of a license that would have allowed you to do an independent implementation of the regular expression APIs from Perl 5?

MS. HURST: Objection. Lacks foundation.

THE COURT: Just a moment.

Well, I think the same foundation applies. Objection

overruled.

Please answer.

THE WITNESS: I'm quite certain that we never sought a license.

BY MR. KAMBER:

Q. Why did you think it was okay to re-implement the Perl 5 regular expression API?

A. Because we've always done things this way. I've been in the profession for a long time, at Sun Microsystems from 1996 to 2004, and before that we have always felt free to re-implement each other's APIs.

Q. At the time that you were at Sun, were you -- what, if any, industry or what, if any, sort of practice were you aware of with respect to re-implementing APIs?

A. You know, as I say, it was all over the place. FORTRAN APIs, which were designed by IBM--

THE COURT: Wait one second. This is -- I'm going to say that is outside the scope of his work at Sun, and at this point, Ms. Hurst would have a good point. So I'm going to sustain the objection on the grounds that this is not his work at Sun. This is a more general statement about what others were doing. That would have to be expert testimony. So I'm sustaining her objection on -- which she made earlier to this particular question.

BY MR. KAMBER:

Q. Dr. Bloch, what did you do after you left Sun?

A. I moved on to Google.

Q. When did you move on to Google?

A. In 2004, just after Java 5 was released, the release that was documented in this exhibit.

Q. And you're pointing to TX984; is that correct?

A. I'm pointing to the third edition, yes.

Q. Now, what did you do at Google when you went there?

A. All things Java.

Q. What do you mean?

A. I had a very sort of wide-ranging role over my time there. I wrote Java APIs at Google for our own internal infrastructure. Google had a file system and it had something called a MapReduce bulk data processing system, and I led a team that implemented Java APIs for these things.

I contributed -- I continued contributing back to this platform, so I actually wrote Java language features as well as Java libraries while I was employed at Google. I actually wrote documentation for the stuff from Java 5 and contributed that back while I was at Google.

You know, I gave talks. I -- once again, I helped junior engineers with their designs, their APIs. You know, too many things to list.

Q. At any time when you were at Google, did you work on

Android?

A. Yes. I worked on Android for approximately one year starting at the very end of 2008 or the very beginning of 2009. I forget which.

Q. Okay. What did do you as a member of the Android team?

A. I worked on these same core libraries -- java.util, java.lang -- and I worked on implementations, independent implementations, of these libraries trying to make them run their best on these mobile devices for which the Android platform was targeted.

Q. Okay. How, if at all, was the work that you did on Android related or specific to the fact that Android was this mobile platform?

A. It was actually quite specific to it because mobile devices have really different constraints from those servers and desktops for which Java 2 SE was written.

So, for example, a server or desktop is plugged in, you have infinite power. You don't have to worry about power consumption. But your cell phone has a little battery and if you use too much power, the battery runs out and that's bad.

So we had to always be conscious of how much power we were consuming. Phones have less memory. They have chips which are called ARM chips, which are far less powerful than Intel chips that run our servers and our desktops, and also it is just different instructions run at different speeds on these two

things. So one has to engineer them specifically to run their best in this constrained environment, and it actually can be quite a challenge.

Q. And remembering back to the code that we had up on the screen, Dr. Bloch, which part of the code was that optimization work being done in? Was it being done in the method declaration or the package or class declaration, or was it being done in the implementing code?

A. Of course it was being done in implementing code, as I hope I showed you. The declarations don't change. The declarations can't change. They are the nexus. They are what allows the caller of a function to call it, but then you write a new implementation that is tuned for the new environment, and that's what we did.

MR. KAMBER: Thank you, Dr. Bloch.

Pass the witness.

THE COURT: Maybe we should take our 15 minute break at this time, give counsel a chance to set up.

Please remember the admonition. No talking about the case. Thank you.

(Proceedings were heard out of presence of the jury:)

THE COURT: Be seated, please. The witness can step outside for a moment.

Can counsel give me the percentage breakdown on the Ellison and Duimovich read-ins?

MR. VAN NEST: Your Honor --

THE COURT: I just need the percentages; in other words, 75 percent, 25 percent.

MR. VAN NEST: We actually were trying to get agreement on the actual minutes.

THE COURT: Well, but, you see, you may be longer than your actual -- give it to me by minutes, but I'm going do it proportionally as to how much time was actually used.

MR. VAN NEST: Google is a hundred percent on Ellison.

THE COURT: All right. One hundred percent. And then on Duimovich?

MR. VAN NEST: Let's see, what's 8 into 13? That's about -- 13 minutes and they had 8 and we had 5. So let's say, what, 67 percent?

MS. HURST: That's fine, Your Honor.

THE COURT: Two thirds, one third.

MR. VAN NEST: Two thirds Oracle, one third Google on that one.

THE COURT: Are you sure?

MR. VAN NEST: Yeah. Yeah. Eight minutes for Oracle, five minutes for Google. Sixty-seven.

MS. HURST: No. We actually disagree with that. I thought he was saying the other way around, Your Honor. I guess we will need some more meet-and-confer on this one.

THE COURT: When we come back out, let me know.

Anything the lawyers -- we have a hanging Exhibit No. 7640. Which one was that?

MR. VAN NEST: That's the book that Mr.-- go ahead.

MR. KAMBER: That's the book, *Effective Java*.

THE COURT: That was the 2008 book?

MR. KAMBER: That's right, Your Honor.

THE COURT: So he didn't really write that while he was working there.

MR. KAMBER: The first edition came out earlier, so he was actually writing and updating it part of the way while he was at Sun.

But, Your Honor, we don't need to move the admission of that book. That's fine.

THE COURT: We're going to leave it out then. That's all I got. Anything you need me for?

MS. ANDERSON: Your Honor, I have a quick logistics question. One of the later witnesses might need to use this easel to draw on. Where does the Court prefer that be located?

THE COURT: It's so small you should pull it around so it's close to the jury. I don't need to see it necessarily, but the jury does. So you need to pull it around kind of right in front of the court reporter.

MS. ANDERSON: Thank you, Your Honor.

THE COURT: Fifteen minutes.

(Recess taken at 9:28 a.m.)

(Proceedings were heard out of presence of the jury:)

THE COURT: May I ask the lawyers a question? The jury asked if we were going to take Friday off next week, and I haven't answered that yet. I told them earlier we would take it off, and I intend to unless the lawyers tell me there's some crisis about taking off next Friday.

MR. VAN NEST: There's no crisis. No, that would be great.

THE COURT: All right. How about on your side?

MR. BICKS: That's fine. I mean, I thought that's where you were going.

THE COURT: Okay. We'll tell them they can count on that so that we --

MR. VAN NEST: You mean count on taking it off or leaving it on?

THE COURT: Taking it off.

MR. VAN NEST: Oh, I thought you were going to add it back --

THE COURT: No, no.

MR. VAN NEST: -- to the schedule. I'm okay either way.

THE COURT: No. I want to give them -- look, they've got a life. I want to let them get back to have one day off next week, next Friday. A week from today we'll be dark.

MR. VAN NEST: Okay.

THE COURT: That's what I'm asking. Is there any crisis with you if we do that?

MR. VAN NEST: No, there's no crisis.

THE COURT: All right. Then we're going to do it. Okay. Let's bring the jury back.

(Pause in proceedings.)

THE COURT: Oh. What's the breakdown on Duimovich?

MR. VAN NEST: I think the bidding is at 60/40, but we don't have a deal yet, but I will give it to you.

Do we have a deal?

THE COURT: Give me the number of lines. That's what I want to know, the number of lines per side. That's an objective thing.

MR. VAN NEST: Well, that's easy for the read-in, but on Duimovich, the lines of what? Oh, of the transcript.

THE COURT: The transcript. You just --

MR. BICKS: Judge, we'll get the answer.

THE COURT: But I need to be able to tell you what the number of minutes everybody has used, and I can't do that until I know how to break down, it looks like, 17 minutes.

MR. BICKS: We're just having people time the video and look at it and we'll get it.

(Proceedings were heard in the presence of the jury:)

THE COURT: During the break, one of you asked Angie if we were going to be taking a week from today off totally.

The answer is yes. So next week we'll be in session Monday, Tuesday, Wednesday, Thursday. We will all take Friday off next week -- what is that? -- the 20th.

And then the following week, we will be in session all five days. And the next day we get off looks like it's -- I'm sorry. Right.

A week after, starting the 23rd, we are in session all five days; and then Monday the 30th is a holiday, Memorial Day, we'll take that off.

So this week we were in session all five days; next week four days, but Friday off; the week after that, all five days again. Okay? We got it? Great.

Ms. Hurst, your turn.

MS. HURST: Thank you, Your Honor.

CROSS-EXAMINATION

BY MS. HURST:

Q. Good morning, Dr. Bloch. My name is Annette Hurst and I represent Oracle.

A. Good morning.

Q. On direct you said there was a beauty in APIs. Do you remember that?

A. I don't, but I'll take your word for it.

Q. And you feel pretty passionately about API design?

A. I do.

Q. It's a topic that you have spoken about extensively over

the years?

A. I have.

Q. Dr. Bloch, I'd like to show you an exhibit that has been marked by the parties as 624.

A. Got it.

Q. Now, this Exhibit 624, do you recognize it?

A. I do.

Q. It's one of your presentations that you gave about designing APIs?

A. Yes.

Q. That's got your name on it and Google's name on it?

A. It does.

MS. HURST: Move the admission of 624.

RIGHT4: No objection, Your Honor.

THE COURT: Received in evidence.

(Trial Exhibit 624 received in evidence)

BY MS. HURST:

Q. Now, this particular presentation, this is one you gave at Javopolis. Am I pronouncing that right?

A. We say Javopolis, but --

Q. Javopolis. In 2005; is that right?

A. Yeah. I believe that was in Antwerp, Belgium.

Q. And who comes to Javopolis?

A. Just programmers, engineers.

Q. Okay. Let's look at page 2 of the document.

A. Got it.

Q. And is that a slide that you wrote?

A. Yes, it is.

Q. In fact, you wrote all of the slides in Exhibit 624; is that correct?

A. Almost certainly, but give me a second.

(Witness examines document.) Yeah. Other than the -- other than the graphics on the templates.

Q. All right. And so you wrote, "APIs can be among a company's greatest assets"; right?

A. I did.

Q. "Can also be among a company's greatest liabilities." And there you explained "Bad APIs result in an unending stream of support calls"; right?

A. I did. Yes.

Q. And I think on direct you said that some APIs are harder than others.

A. Harder to --

Q. Harder to write than others?

A. Yes, some are harder to write than others.

Q. And why are some APIs harder to write than others?

A. Because of the complexity of figuring out how best to express what it is that the programmer wants done. Some tasks are more complicated than others.

Q. All right.

And one of the other things you said on this slide is that "Public APIs are forever. One chance to get it right." That's what -- and I think you said something like that on direct here today as well; right?

A. No. I did not say that on direct today.

Q. Okay. Well, this is -- you presented this; true?

A. That is true.

Q. And you believed that; right?

A. I believed it at the time more than I do now. In the interim, Java has taken a few more chances at some of its APIs. You end up with a little bit of a mess when you have multiple APIs trying to do the same things, but you can try and do it.

Q. Let me show you, sir, Exhibit 877, as well. Do you recognize that exhibit?

A. I do.

Q. And is that a document that you authored in or about September 2008?

A. Not really. I mean, it's derivative. It comes from the -- what do you call that? -- the abstract of a talk that I gave at OOPSLA in 2004 on the same topic.

Q. You published this particular version of it --

A. I didn't publish --

Q. -- Exhibit 877, on or about September 7, 2008; true?

A. Technically, no. InfoCue published it.

Q. Did you authorize the publication by InfoCue in or about

September of 2008 of this document, sir?
A. Of course.
MS. HURST: Move the admission of 877, Your Honor.
MR. KAMBER: No objection.
THE COURT: Thank you. Received.
(Trial Exhibit 877 received in evidence)
BY MS. HURST:
Q. All right. In 2008, Dr. Bloch -- let's look at the third bolded line here, Trudy -- (reading)
"Public APIs like diamonds are forever. You have one chance to get it right so give it your best."
That was what was published in Exhibit 877 when you authorized that publication; is that right, Dr. Bloch?
A. It is, and it's good advice even if it's not technically a hundred percent correct.
Q. All right. And, again, here in this document, you published: (reading)
"APIs can be among" -- or pardon me.
"InfoCue-published APIs can be among your greatest assets or liabilities."
And that matches what you said in your other presentation at Javopolis; right?
A. Close, although you'll notice I say "your" rather than "your company's" because I realized that I kind of got it wrong the first time. It applies to any organization that supports

an API, whether it's an open source project or a company or whatever.

Q. The bottom line is, you believe API design is important?

A. Absolutely.

Q. And if we can, Trudy, go back to Exhibit 624.

Let's look at page 15, Dr. Bloch. Now, this is another one of the slides that you presented at Javopolis in 2005; is that right?

A. That is correct.

Q. And your advice was, as reflected on this slide, "Implementation should not impact API." Did I read that correctly?

A. Absolutely.

Q. And let's look, Trudy, at page 47.

And this was your concluding slide at Javopolis in 2005, Dr. Bloch; is that right?

A. That's right.

Q. And you included on that slide a bullet point "API design is a noble and rewarding craft." Did I read that correctly?

A. You did.

Q. You also said that API design is tough; true?

A. Yeah. And, I mean, when I spoke to the bullet.

Q. Just yes or no, Dr. Bloch.

A. That is what I wrote.

Q. Is that what you said, that it was tough?

A. I said it's tough to do it well.

Q. Because it's -- strike that.

And, in fact, if we look back at page 8 -- pardon me -- Exhibit 877, this abstract, you called it "Bumper sticker API design"; is that right?

A. Yes, I did.

Q. All right.

A. It was an homage to another computer scientist.

Q. And if we look at the very end of your abstract, which is on page 2 of the exhibit -- in the very last bullet, Trudy, just above there. Sorry. Not the comments. There we go -- Bumper sticker API design, you called it an art not a science; correct?

A. Yes.

Q. You've actually had people tell you that your APIs have changed their lives?

A. I think it was said facetiously but, yes.

Q. But you thought there was no higher compliment --

A. Yes.

Q. -- that someone could pay to an API designer; isn't that true?

A. Absolutely.

Q. All right. Let's go to a different topic, Dr. Bloch.

I'm going to show you Exhibit 358 do you recognize Exhibit 358?

A. I do.

Q. And that's an email string between you and Jesse Wilson at Google on December 7, 2009; is that right?

A. Yes, it is.

MS. HURST: Move the admission of Exhibit 358.

RIGHT4: No objection.

THE COURT: Thank you. Received.

(Trial Exhibit 358 received in evidence)

BY MS. HURST:

Q. And who was Mr. Wilson at this time, Dr. Bloch?

A. He was an engineer on the core libraries group of the Android project.

Q. All right. And Mr. Wilson talked about a hurdle of OpenJDK licensing under the GPL plus Classpath exception. Do you see that? It's in the second paragraph --

A. Yeah.

Q. -- of his email. Do you see that?

A. I do.

Q. And you responded to his email Jesse: (reading)

"Ooh, you put your hand into the buzz saw. Josh."

That was your response, wasn't it?

A. Yes.

Q. And you wrote that because at the time you understood that using OpenJDK in Android was a verboten topic at Google; correct?

A. I have said that at one point in my life, but on further reflection, I think I misspoke.

MS. HURST: Your Honor, permission to play a clip from the July 8th deposition at 206:16 to 207:9.

THE COURT: What is the actual date? What year?

MS. HURST: Oh, Your Honor, I apologize. It is July 8th, 2011.

THE COURT: All right. Any objection?

MS. HURST: I'm sorry, Your Honor. 206:3 to 206:21.

THE COURT: All right. Any objection?

MR. KAMBER: No, Your Honor.

THE COURT: All right. Please play it.

(Whereupon, the video was played for the jury)

BY MS. HURST:

Q. You were under oath when you gave that testimony, Dr. Bloch?

A. I was, and I agree with every bit of it, except the verboten topic. You probably saw the long pausing because I thought, well, it's really controversial rather than verboten. That's why I said I misspoke.

Q. Did you ever write down a correction for that testimony and send it in to the court reporter and the lawyers in the case?

A. No, I did not.

Q. All right. Sir, let me refer you to Exhibit 4027, which

Mr. Kamber asked you about. It should be up there. I can help find it, if you like.

A. Got it.

Q. And you said you were familiar with this book when it was published by Sun?

A. I was actually familiar with it before it was published.

Q. All right. Let's look at page 7 of the exhibit. And you see there's a paragraph there -- well, first of all, at the top it says "Copyright 1996, Sun Microsystems, Inc." Did I read that correctly?

A. You did.

Q. And there's a paragraph here starting with: (reading)

"Sun Microsystems hereby grants you a fully paid nonexclusive, nontransferable, perpetual, worldwide limited license without the right to sublicense."

I'm not going to keep reading. Do you see that paragraph, sir?

A. I do.

Q. And that paragraph was in this book when it was published; isn't that true?

A. I don't know.

Q. You never looked at it?

A. No.

MS. HURST: No further questions.

THE COURT: Thank you.

Anything more?

MR. KAMBER: No, thank you, Your Honor.

THE COURT: May Dr. Bloch be excused?

MS. HURST: Yes, Your Honor.

THE COURT: Discharged from the subpoena?

MR. KAMBER: Yes, Your Honor.

THE COURT: Okay. You're free to go. Thank you.

THE WITNESS: Thank you.

THE COURT: Have a great day.

(Witness excused.)

THE COURT: And let's have the lawyers come up and remove the exhibits so we can move to the next witness.

And, meanwhile, who will be the next witness?

MR. VAN NEST: We're going to have a short prior testimony video, and Mr. Mullen will introduce that.

THE COURT: All right. So, Mr. Mullen, come up here but I want to -- I don't want you to interfere -- I don't want the cleanup to interfere, but just come up here and be in the ready position.

MR. MULLEN: I'll pause, Your Honor.

THE COURT: You'll do what?

MR. MULLEN: I'll pause.

THE COURT: All right. Mr. Mullen, who is our next witness?

MR. MULLEN: Good morning, ladies and gentlemen. My

name is Reed Mullen. I'm one of the lawyers representing Google in this case.

The next witness we're going to here from is by video. It's Mr. Donald Smith. His sworn testimony was taken on November 20, 2015, and at the time of his deposition he was an employee of Oracle.

THE COURT: All right. Don't play it yet until we get all this stuff out of the way; and then once we have the jury's undivided attention, hit the button.

(Pause in proceedings.)

THE COURT: Roll the tape.

(Whereupon, the video deposition of Donald Smith was played for the jury not reported.)

MR. VAN NEST: We need to start that again. Excuse us, Your Honor.

THE COURT: What's the issue?

MR. VAN NEST: Just the system needs to be switched, and madame clerk is going to do that right now.

(Whereupon, the video deposition of Donald Smith was played for the jury not reported.)

MR. VAN NEST: That concludes the presentation, Your Honor.

THE COURT: All right.

MR. VAN NEST: We're ready to call our next witness.

THE COURT: Please do.

MR. VAN NEST: Google will call Mr. Simon Phipps, and Mr. Kwun will present his testimony.

THE COURT: Okay.

(Pause in proceedings.)

THE COURT: Would one of you hand up to me the order I did in this case on Mr. Phipps? I just want to have it in front of me, or maybe my law clerk can go get it. My law clerk will get it.

Angie, would you hand it to me, please.

Are you Mr. Phipps?

THE WITNESS: I am, yes.

THE COURT: All right. Welcome to the court and please raise your right hand and take an oath to tell the truth.

SIMON PHIPPS, DEFENDANT WITNESS, SWORN

THE CLERK: Please state your name for the court and spell your last name for the record.

THE WITNESS: My name is Simon Phipps, and my last name is spelled P-H-I-P-P-S.

THE COURT: All right. Great.

First question.

DIRECT EXAMINATION

BY MR. KWUN

Q. Good morning, Mr. Phipps.

Can you tell us, please, a little bit about your

educational background?

A. I have a degree in electronic engineering from the University of Southhampton in the UK. I got that in 1982. And that was a course that covered both electronics and computer system programming.

Q. And where do you live?

A. I live in Southhampton in the UK right down on the south.

Q. Have you ever worked as a computer programmer?

A. I have, yes. I worked -- when I left college, I went and worked for a company called Burroughs Machines, and I worked as a system programmer on their medium systems range in COBOL, and then later on in Pascal on their CTOS.

Q. What years did you work at Burroughs?

A. I -- I joined Burroughs when I left college and I left in 1990.

Q. And when you left Burroughs, where did you go?

A. I went to IBM United Kingdom Laboratories, which is based in Hursley Park near Winchester in the UK.

Q. And what was your job title when you joined IBM?

A. Quality assurance specialist.

Q. Did there come a time when your job role changed?

A. Yes. IBM rolled the quality assurance function into the product groups, and I joined the person-to-person collaborative conferencing team.

Q. What was your role on the person-to-person collaborative

conferencing team?

A. So I handled all of the outward-facing tasks for the community there. So that was things like speaking at conferences, liaising with our collaborative partners -- I worked with Apple -- and working on a standards group over video standards.

Q. How long did you have that role?

A. I kept that role until our team was reassigned in 1995, and I was one of the founders of IBM's Java Technology Center.

Q. So 1959, if you look at our -- we have a timeline up here, 1995, that would have been around when Java released the -- excuse me -- when Sun released the Java programming language; is that correct?

A. Yes. Java was released in the spring of 1995, and our team was assigned to evaluate and support Java in the summer of 1995.

Q. Did you have a title on the Java team at IBM?

A. Yes. I was IBM's chief Java evangelist.

Q. And what did you do as the chief Java evangelist at IBM?

A. Again, I did all of the outward-facing tasks for IBM in relation to Java. So, for example, I spoke to the media. There's an article about me in ***Wired*** magazine. I attended the JavaOne conference on behalf of IBM. I gave lectures at the college. Anything that involved representing IBM in public on Java.

Q. How long were you the chief Java evangelist at IBM?
A. I kept that job title laterally in conjunction with another job title until I left in 2000.
Q. What did you do when you left IBM in 2000?
A. I joined Sun Microsystems in 2000.
Q. What was your title when you joined Sun?
A. When I joined Sun, I was the chief technology evangelist in their software group.
Q. What was your -- what were your job duties as the chief technology evangelist?
A. I was the person -- I was part of a team of people developing our messaging around service-oriented architectures and XML products, and I communicated that at conferences and in the media.
Q. Did there come a time while you were at Sun that your title changed?
A. Yes. When Jonathan Schwartz became CEO at Sun, I had a conversation with him about how important it would be to have an executive handling open source for the company, and he agreed that I should be Sun's chief open source officer.
Q. So what year was that in?
A. I don't remember precisely, but it was the end of 2004, beginning of 2005.
Q. And what did -- what was your role as the chief open source officer at Sun?

A. There were both internal-facing and outward-facing roles. I ran a team of people who provided internal consulting services on open source community dynamics, open source practice, open-source licensing.

And I also worked with the legal team to work on Sun's messaging around open source. And then I also represented Sun in public at conferences, like the open source convention at -- I spoke at JavaOne. I spoke at conferences in Australia and in the Far East.

And I also represented Sun in the media and also represented Sun's positions on the blogging website, blogsun.com, which I helped start.

Q. How long did you hold the position at Sun of being chief open source officer?

A. I held that position until I left the company, which was 2010.

Q. When you left in 2010, was that before or after Sun was acquired by Oracle?

A. That was after Sun was acquired by Oracle and immediately contingent on the acquisition of the part of Sun that I actually worked for, which was Sun Microsystems Limited.

Q. Why did you leave Sun?

A. I left because Oracle didn't need a chief open source officer.

Q. And what do you do now?

A. Now I run a company that provides management consulting on open source community and practical issues and also on digital rights in the UK.

Q. Have you been retained by Google as a consultant in this case?

A. I have, yes.

Q. Are you being paid for your testimony here today?

A. No. This is all my own words.

Q. Are you being paid for the time you've taken to prepare for this testimony?

A. Yes. Google are compensating me for my time.

Q. Mr. Phipps, can you please explain what the term "open source" means?

A. "Open source" refers to a copyright licensing regime where copyright licenses give to any end user or software developer the freedom to use software for any purpose in its source code or object code form, the freedom to make whatever changes they want to to the program, and the freedom to pass on the original or the modified version to anyone that they want to without needing to seek further permission from anybody and without any restrictions on the field in which they may use the software.

Q. You mentioned a couple of terms in your answer there. Can you explain what "source code" is?

A. "Source code" is the -- the human-readable version of a program that's commonly used by computer programmers to write

their computer programs.

Q. How about "object code"? What's "object code"?

A. "Object code" describes the computer-readable version of a computer program, and you typically convert a -- the source code into object code. It's -- there are a variety of ways that that happens, but that's the typical way that it happens.

Q. To what extent, if at all, can open source code be used in commercial products that are sold for profit?

A. Oh, it's fundamental to the ideal of open source that you should be able to use open source software for commercial purposes. It's written right there in the open source definition.

Q. Are you familiar with a product called OpenJDK?

A. I am, yes.

Q. What is OpenJDK?

A. OpenJDK is the name used to describe Sun's Java SE platform when it's made available under an open source license.

Q. And in that name OpenJDK, can you tell us what the JDK part stands for?

A. JDK stands for Java development kit, and it was kind of the colloquial term that developers have been using from the combining of the Java platform to refer to Java and the tools they needed to work on it.

Q. When did Sun first announce OpenJDK?

A. So the idea of open sourcing Java was first announced in

JavaOne in 2006, which I think was in May, and then the details were painted in November of 2006, and then the code was mostly made available in the spring of 2007, around the time of JavaOne again.

Q. Who was it that announced open -- excuse me -- the open sourcing of Java SE at JavaOne?

A. That was announced by Jonathan Schwartz.

MR. KWUN: Your Honor, can I approach the witness?

THE COURT: Yes.

BY MR. KWUN:

Q. Mr. Phipps, I've handed you a series of exhibits, but if you could find Exhibit 7717. They should be in numerical order in there.

A. (Witness examines document.)

Q. Let me know if you recognize Exhibit 7717.

A. I do.

Q. What is Exhibit 7717?

A. This is an article that I wrote in August 2006 that was posted on Sun's corporate blogging website, blogsun.com.

MR. KWUN: Your Honor, we would offer 7717 into evidence.

LEFT5: No objection, Your Honor.

THE COURT: Thank you. Received.

(Trial Exhibit 7717 received in evidence)

BY MR. KWUN

Q. If we could take a look at the first paragraph. Is this a blog post, first of all?

A. Yes.

Q. If we could take a look at the first paragraph of this blog post. At the very end of this you say: (reading)

"The Java platform is more a market maker than a direct revenue generator."

Do you see that?

A. I do.

Q. What did you mean by that?

A. Well, this article was written in the aftermath of the announcement that Java would be open source but before the details became apparent, and people were commenting that this was going to result in a loss of revenue for Sun.

And I was explaining that the point of Java was not so much the direct generation of revenue, although that was useful, but rather that it was creating a marketplace in which Sun was able to sell its products and skills and the reputation that was a door opener for our sales force, and that was by far the most important dimension of Java for Sun in generating opportunity.

Q. Mr. Phipps, are you generally familiar with the 37 Java SE API packages that are at issue in this case?

A. I am.

Q. Were those packages part of OpenJDK?

A. Yes.

Q. If you could take a look at Exhibit 7722 and let me know whether you recognize that.

A. Oh, yes.

Q. What is Exhibit 7722?

A. This is a generation of the open source FAQ that my team and the Java marketing team collaboratively produced for open source Java.

Q. What does "FAQ" stand for?

A. "FAQ" stands for frequently asked questions. In the open source community, because there's a lot of mailing lists and forums, people ask all the same questions all the time and it gets tiresome answering them, and so we collected together all the most frequently asked questions and the questions we thought people would want answers to, and then our teams agreed Sun's party line on what the answer to the questions would be. So it was the answers to frequently asked questions in the community.

Q. Where was this published?

A. This was published on Sun's main website at Sun.com.

MR. KWUN: Your Honor, we'd offer 7722 into evidence.

LEFT5: No objection.

THE COURT: Thank you. Received.

(Trial Exhibit 7722 received in evidence)

BY MR. KWUN:

Q. Mr. Phipps, who was the intended audience for these FAQs?

A. These were written in general for software developers, and specifically they were written to answer the concerns of the open source community and of Java developers who were interested in open source.

Q. If you could turn to page 23 of the exhibit, there is a heading down towards the bottom that says "Open Source Communities and Java." Do you see that?

A. I do.

Q. And the first question underneath that asks: (reading)

"Have you been engaging with the non-Sun Java SE platform communities such as Apache Harmony, GNU Classpath and Cafe?"

Do you see that answer?

A. I do.

Q. What was Apache Harmony?

A. Apache Harmony was a project at the Apache Software Foundation to create an independent implementation of the Java SE platform under Apache's preferred open source license.

Q. Did Apache Harmony include the Java SE API packages?

A. Yes.

Q. And what was GNU Classpath?

A. GNU Classpath was a project at the GNU project to create an implementation of the -- just the class libraries for

Java SE for use in conjunction with an open source virtual machine of the users choice, in many cases Cafe, which is also mentioned in that answer.

Q. Class libraries, is that the same as API packages?

A. It's a little bit different. A class library is the implementation of the code. The API is the way that the developer accesses that implementation.

Q. The GNU Classpath class libraries, did that include the Java SE declarations?

A. It did, yes.

Q. The answer beneath that question says that: (reading)

"The Java developer ecosystem has a lot of very smart, experienced, community savvy people who are passionate about the platform and eager to help."

Do you see that?

A. I do.

Q. And then it says: (reading)

"We've been in contact with these projects over the last few years, but in much more regular contact during the last few months."

To what extent, if at all, were you involved in that contact?

A. That was something that I frequently did. The people named there -- Geir and Dalibor and Mark -- were all people that I maintained relationships with, and that I still know as

friends actually.

Q. Did you contact Geir Magnusson of Apache Harmony on behalf of Sun?

A. I did, yes.

Q. And did you contact Mark Wielaard of the GNU Classpath project on behalf of Sun?

A. Yes. I traveled specially to meet him.

Q. And at the end of that question, it says -- or excuse me -- the end of that answer it says: (reading)

"We are very grateful to all of these people for their help and advice they have so generously and graciously offered."

Do you see that?

A. I do.

Q. Was that accurate?

A. Yes.

Q. The next question below that asks: (reading)

"Are you planning to work with these communities directly?"

Do you see that?

A. I do.

Q. And the answer which actually continues on to the next page a little bit -- and if we could get that up as well, that would be great -- but the answer says: (reading)

"The Java ecosystem can support multiple

implementations."

Was that an accurate statement?

A. Yes.

Q. Was that Sun's position at the time?

A. It was.

Q. It goes on to say: (reading)

"Choice and differentiation keeps both commercial and open source implementations on their toes, and we're not expecting any of the existing open source Java SE or Java ME implementation communities to," quote, "'close up shop,'" unquote, "now that the JDK and Java ME implementations have been open sourced."

And then it says: (reading)

"It wouldn't be good for Java technology if they did," exclamation point, "indeed."

Was that an accurate statement?

A. Yes.

Q. And was it Sun's position at the time that it wouldn't be good for Java technology if those communities closed up shop?

A. That was the position, yes.

Q. Did anyone at Sun ever ask you to take action to stop Apache from distributing Apache Harmony?

A. No.

Q. To your knowledge, did Sun ever tell Apache to cease its distribution of Apache Harmony?

A. No.
Q. Did anyone at Sun ever ask you to take action to stop GNU from distributing the GNU Classpath project?
A. No.
Q. And to your knowledge, did Sun ever tell GNU to cease its distribution of the GNU Classpath project?
A. No.
Q. If you could turn to page 13 of Exhibit 7722. In the middle of the page, there is a question that asks: (reading)

"Can someone create an implementation that isn't compatible with the Java specification using this code?"

Do you see that?
A. I do.
Q. And the answer there is yes; is that right?
A. That is correct.
Q. There's also a reference -- it says: (reading)

"In addition, they cannot label that implementation with the Java-compatible or Java-powered brand and logo."

What does that mean?
A. Well, one of the core propositions of the Java platform to software developers was the ability to write a computer program and expect it to behave in the same way regardless of which platform that you ran it on. And so Sun had a very well-defended trademark or set of trademarks that were used to give an indication to software developers that the platform

they were targeting was indeed a compatible platform. That included that familiar coffee cup logo with the steam over the top, and it also included the words that you see in the answer here.

And we fully anticipated that anybody who tried to create an incompatible version of Java and call it Java would be unable to use the brand, and that that would warn away software developers from trusting that implementation.

Q. Mr. Phipps, the question below that on that page asks whether someone can create software that doesn't even implement Java but uses pieces of the JDK. Do you see that?

A. I do.

Q. And the answer says, "There are no limitations," and then goes on to explain that you would need to comply with the license terms. Do you see that?

A. I do.

Q. In that answer, is that supposed to include the possibility that someone could use some but not all of the class libraries from OpenJDK in their product?

A. Yes, it is. In fact, we explicitly considered that possibility when we were discussing that answer.

Q. When you say "we," who are you referring to?

A. That's the overall Java team that was working on OpenJDK that included engineering, marketing, business, and our open source group.

Q. Mr. Phipps, what license applied to the API packages in OpenJDK?

A. We chose the GNU General Public License Version 2, and many of the files were additionally given the permissions that come with something called the Classpath exception.

Q. If you could turn to page 11 and 12 of the Exhibit 722, there are a couple of questions.

And if we could get up on the screen the question that says "What is the Classpath exception," and the answer; and, "Why do you need the Classpath exception," and the answer which continues on to the next page.

What involvement, if any, did you have in the decision by Sun to license the API packages and OpenJDK under this license?

A. I was a part of the collaborative team that made the decision to choose the license.

Q. And you've mentioned that the license was the General Public License Version 2 with the Classpath exception. In these answers, there's a reference to GPLV2. Is that the General Public License Version 2 that you were talking about?

A. Yes, that is.

Q. And why did Sun choose the GPLV2 with the Classpath exception as the license for the OpenJDK API libraries?

A. To advance that, you have to go back to our reason for making Java open source in the first place. The -- we were facing a problem with Java. The open source community was

becoming a very dominant way of software development, and we discovered that open source developers were avoiding using the Java language.

They were doing that for two reasons. It was because Java was very hard to install on their preferred operating systems, notably the Linux range of operating systems, and it was also because Java was available under licensing terms that they didn't understand that were very complicated and that were proprietary, which are all red flags to the open source community.

And so we wanted to take action that would gain the trust of the open source community. So we were looking through open source licenses and the GNU GPL had all of the attributes that met our business objectives, but it was also a standard license that we knew the community already accepted. And by choosing that standard license, we were able to gain the trust of the community, and that went on to actually be demonstrated to be a good decision because we did indeed have developers join us very quickly when we made the announcement.

Q. And what was the thinking behind the Classpath exception that you applied?

A. Well, the Classpath exception arose out of the earlier GNU Classpath project which had been implementing class libraries for Java SE, and it -- within that open source community, the developers felt there was a risk that people would be inhibited

from developing for their platform by the reciprocal requirements of the GNU general public license.

The GNU general public license gives you a license contingent on you, as a reciprocal move, making your work available under the same license that you received the code under. And the GNU Classpath developers believed that Java developers would be very worried about that term in the GNU general public license, and so they wrote a form of words that made an exception granting extra permissions to software developers not to need to comply with that aspect of the GNU general public license under the most common circumstances.

And we had just the same problem in terms of reality and perception, and rather than crafting our own words, which would be mistrusted by the open source community, we decided to use exactly the same language that the open source community had already used in the equivalent situation in order to secure their trust. And, again, that turned out to be a good decision because people didn't even ask us questions about it.

Q. Take a look at Trial Exhibit 7754 and let me know whether you recognize that document.

A. Okay. Yes.

Q. What is Exhibit 7754?

A. So this is a copy of the GNU General Public License Version 2 and of the Classpath exception as was posted on the Sun's OpenJDK source code forge.

MR. KWON: Your Honor, we offer 7754 into evidence.

MS. SIMPSON: No objection, Your Honor.

THE COURT: Received.

(Trial Exhibit 7754 received in evidence)

BY MR. KWON:

Q. If you could turn to the last page of this exhibit, page 6, there's a reference to the Classpath exception. Is this the Classpath exception you've been testifying about?

A. Yes. It is.

Q. And is the -- which paragraphs in here are actually the Classpath exception?

MS. SIMPSON: Your Honor, objection. I believe we are verging into expert testimony on licensing terms.

THE COURT: Sustained. You're talking present tense. You've got to be past tense and establish that he actually looked at this back then.

BY MR. KWON:

Q. Mr. Phipps, were you involved in the selection of this license at the time that OpenJDK was released?

A. Yes.

Q. Did you look at this document at that time or about the time that OpenJDK was released?

A. I did, yes.

Q. Based on your knowledge and your recollection of having worked on the release of OpenJDK, can you tell us which

paragraphs in here are the Classpath exception?

THE COURT: No. Which paragraphs back then --

MR. KWON: Back then were the Classpath exception.

THE COURT: -- did he understand be the Classpath, that would be okay.

MS. SIMPSON: Your Honor?

THE COURT: Yes?

MS. SIMPSON: Just an additional objection, we're getting into the area of privileged territory based on legal advice.

MR. KWON: Your Honor, I just want him to identify which paragraphs are the Classpath exception. I'm not planning on going into any detail.

THE COURT: All right. You can do that much.

MR. KWON: Thank you.

THE COURT: I don't think that will -- but we're not going to reveal attorney-client.

But if you think it will reveal attorney-client for you to answer the question, then don't answer. But can you answer without revealing attorney-client?

THE WITNESS: Absolutely. This is a public document.

THE COURT: All right. Go ahead.

BY MR. KWON:

Q. Mr. Phipps, which of these paragraphs are the Classpath exception?

A. They are the -- the indented paragraphs.

Q. So that would be the one starting with "linking" and the one starting "as a special exception"?

A. Yes.

Q. Mr. Phipps, can you please take a look at Exhibit 7578. Let me know whether you recognize that exhibit.

A. I do.

Q. What is Exhibit 7578?

A. This is an email that I sent to one of Sun's internal mailing lists on November the 8th, 2005.

MR. KWON: Your Honor, we would offer 7578 into evidence.

MS. SIMPSON: Objection. We object to this based on the order that you issued with respect to Mr. Phipps.

THE COURT: All right. May I see it?

(Pause in proceedings.)

THE COURT: Well, this is a historical about when he was at Sun; right?

MS. SIMPSON: Your Honor, it's a third party communicating to another third party, and your order talks specifically about his experiences at Sun.

MR. KWON: Your Honor, it goes to Sun's knowledge.

THE COURT: It says from him to somebody else. Remind me who Tim Ellison is.

MR. KWON: Tim Ellison was an IBM employee.

THE COURT: All right. Did you receive this, Mr. Phipps, while you were at Sun?

THE WITNESS: Oh, yes.

THE COURT: Did you read it when you received it.

THE WITNESS: Yes. Yes.

THE COURT: Did you use this information in some way as part of your work at Sun?

THE WITNESS: Yes.

THE COURT: I think that fits enough within the parameters to allow it. So the objection is overruled, but it's still not yet in evidence.

Have you moved it in evidence?

MR. KWON: I did move it in evidence, Your Honor.

THE COURT: All right. It will be received, though, only for purposes of the information that was known to Sun, not necessarily for the truth of it because it attaches that Ellison email.

MR. KWON: Yes, Your Honor.

THE COURT: With that limitation, it's received.

(Trial Exhibit 7578 received in evidence)

BY MR. KWON:

Q. Why did you send this -- first of all, what is the open source at Sun.com email address?

A. Sun had many internal mailing lists so that all of the employees who were interested in different topics could see

what was happening in the world and inside the company. And open source at Sun.com was a mailing list that anyone in the company was free to join that -- where we would share information about open source topics in general, both Sun's activities and also the activities in our marketplaces.

Q. And in this November 8th, 2005, email, what information were you sharing with the open source mailing list?

A. Well, we were well aware that Apache was working -- had said they were going to work on the Harmony project, and I sent this to the mailing list to show people that it had moved beyond the idea of just a project creating a job recipe implementation, and there had actually been a significant contribution of code to bootstrap or to get started that project at Apache.

MR. KWON: Thank you, Your Honor. I pass the witness.

THE COURT: All right. Let's go to cross-examination. Ms. Simpson are you ready, or do you need time to set up?

MS. SIMPSON: I think I'm pretty much ready, Your Honor.

THE COURT: All right.

CROSS-EXAMINATION

BY MS. SIMPSON:

Q. Good morning. I'd like to reintroduce myself. My name is Lisa Simpson. I'm here from Orrick on behalf of Oracle.

Good morning, Mr. Phipps. How are you?

A. Coping so far, thank you.

THE COURT: May I suggest that you turn the microphone closer to your voice?

MS. SIMPSON: Is that better?

THE COURT: That is better. Thank you.

BY MS. SIMPSON:

Q. Mr. Phipps, I'd like to start with a document you were looking at on direct. It's document Trial Exhibit 7722.

A. 7722. Okay.

Q. Do you have that in front of you?

A. Yep.

Q. And you indicated that this was an FAQ relating to open source, did you not?

A. Yes.

Q. And you indicated that this stated Sun's position with respect to the open source -- the releasing of OpenJDK; is that right?

A. Yes.

Q. And I'd like to direct you to a couple of paragraphs in this document. If you look at the page marked with Bates 8122.

A. Eight one -- I don't see that number.

Q. I'm looking at the Bates number. It's 3 of 24.

A. Okay.

THE COURT: Angie, I think we need the document on the screen. There we go.

THE WITNESS: Okay.

BY MS. SIMPSON:

Q. I'm looking at the top of the document, Mr. Phipps. And this document makes clear that although Sun was releasing OpenJDK, it was also going to be maintaining and distributing its commercial licenses; correct?

A. Yes.

Q. And that's at the top of the page, the Q and A, "Will Sun continue to distribute its commercial implementation of the JDK?"

A. Yes.

Q. And the same was true for ME; right, Mr. Phipps?

A. Yes.

Q. If you turn to the next page. And that one is at the bottom of page 5 of 24. Question: (reading)

"Will Sun continue to ship commercial implementations?"

And this is in regard to ME?

A. Yes.

Q. And Sun's going to continue to offer commercial licenses even though it's offering OpenJDK; correct?

A. Correct.

Q. And on this same page, Mr. Phipps, doesn't this question and answer also indicate that Sun is -- while it's open sourcing its ME, it's going -- it's Java ME, it's also going to

open source a couple versions of Sun Java Wireless Client. Do you see that near the top of the page where it says "What is Sun open sourcing in Java SE?"

A. I see the question.

Q. Do you see it?

A. I can -- yeah, I can see the question. Which part of the answer? It's quite a long answer.

Q. Right. And it indicates that Sun is going to release Sun Java Wireless Client, and it says it's the next generation version of the platform that currently enables rich mobile data services in over 1.5 million handsets. Do you see that?

A. I do.

Q. Then it says: (reading)

"Later this year Sun will release its advanced operating system, or OS, phone implementation based on the connected device configuration, the CDC specification."

Do you see that?

A. I do.

Q. And if you go down a little bit on the page, it defines what those terms mean. And if you look, it says: (reading)

"What is the advanced operating system phone implementation?"

Do you see that?

A. I do.

Q. And it says: (reading)

"The advanced operating system phone implementation is a Java Runtime designed to run on operating systems targeting advanced mobile devices like smartphones and set top boxes."

Do you see that?

A. Uh-huh.

Q. And this is the official statement of Sun; right?

A. This was a -- as I say, this was a documented created collaboratively by the team, and it was our party line if we were asked a question.

Q. Right. You testified on direct that it was an official statement of Sun.

A. Yeah.

Q. I'd like to turn to page 9 of 24, which is on 8122 is the Bates number, if that helps you. I'm sorry. 8128.

And this document also indicates -- this Sun Q and A also indicates the importance of compatibility for Java, does it not?

A. It does.

Q. And compatibility was a very important aspect of the Java platform, was it not?

A. Yes.

Q. And you've written about that in your blogs, about how important compatibility was to Java; right?

A. Yes.

Q. And what does "compatibility" mean?
A. So "compatibility" means that a computer programmer can reasonably expect that their program will behave the same way regardless of which compatible Java implementation they run the program on.
Q. And Sun often referred to that as "write once, run anywhere"; correct?
A. That was in our aspirational goal, yes.
Q. And if you also look -- now I'm still on page 9 of 24.
A. Okay.
Q. You see where it says "Do you think anyone will fork the JDK?"
A. Yes.
Q. Can you explain to the jury what a "fork" is?
A. So when this document was written, "fork" meant that a group of software developers would take some open source code and would develop it independently of the original developers of the code, rather than collaborating with them over the same version. And so the result would be that there would be two versions that were not kept in sync with each other.
Q. And this document, this FAQ that Sun prepared, it expressly states in this Q and A that: (reading)

> "Broad distribution of incompatible forks is potentially a danger since such forks could damage the 'write once, run anywhere' compatibility value of the Java

platform."

Correct?

A. That's correct.

Q. And it also says that: (reading)

"Again, the Java technology compatibility promise is so central to the value of the platform that you would want to protect that."

Correct?

A. Yes.

Q. One more thing on this page, Mr. Phipps. This document also says that by offering OpenJDK under the GPL, that doing so makes forks, proprietary forks -- that would be incompatible forks -- less likely, does it not?

A. Could you direct me to that? It's quite -- that's, again, quite a long paragraph.

Q. It's about -- halfway down the question is "So what about compatibility," and it's the fist indent under the answer.

A. Okay. (Witness examines document.)

Q. It says, "License. GPL makes proprietary forks less likely."

A. All right. Okay. Yes.

Q. That's true, isn't it?

A. So that's what the document says, yes.

Q. I'd like to turn to the end of the document. This is at page 15 of 24.

A. Okay.

Q. And I'm looking at the bottom of the page: (reading)

"How can Sun have other licenses that bear on the open source code base? Isn't that no longer open source?"

Do you see that?

A. I do.

Q. Can you read what Sun's response was to that question?

A. Okay. The answer says: (reading)

"Because Sun owns the copyright for the open source code base, Sun is able to license each copy of this code base distributed by Sun under any license, including a commercial software license. This right is inherent in copyright law. Several free and open source communities exhibit this behavior."

Q. And as chief open source officer at Sun, Mr. Phipps, did you talk regularly to open source individuals at Google?

A. It was very infrequent.

Q. Did you speak with Chris DiBona, the director of open source at Google?

A. Around about the time we're talking about here, I did not frequently speak to Chris, no.

Q. Did you speak to him later in your job duties, like in 2010?

A. Yes. Later on in the -- so I typically met Chris at community events so -- particularly at the open source

convention.

Q. And what about Daniel Berlin at Google, did you speak to him? He was in charge of open-source licensing and engineering.

A. It doesn't ring any bells.

Q. You didn't speak to him at Google?

A. Doesn't ring any bells, no.

Q. And although you had these interactions with Google, Mr. Phipps, Google never consulted you about using OpenJDK for Android in the 2005 to 2007 time frame, did they?

A. No.

Q. And no one at Google ever told you in the 2005 to 2007 time frame that Google was considering using the Java APIs from OpenJDK for Android did they?

A. No. I wouldn't expect them to at that time.

Q. So you have no knowledge from 2007 about whether Google thought it could use OpenJDK for Android, do you?

A. I've no insight into their views at that time.

Q. Mr. Phipps, you testified about GNU Classpath on direct. Do you remember that testimony? You don't actually know about all the details with respect to Sun's interactions with GNU; isn't that right?

A. I picked up the relationship with GNU. So I had a predecessor at Sun, Denise Cooper, who maintained the relationships with the free software foundation and the GNU

project -- and, incidentally, helped to start the Harmony project -- and she looked after all the relationships up to the point where I took over that role.

Q. So you're not aware of enforcement efforts that Sun may have taken against GNU prior to your role as chief open source officer; correct?

A. None were brought to my attention, no.

Q. So you're not aware of an effort Sun made to cease distribution of Java by a Korean entity who claimed to be using GNU Classpath libraries in their products; correct?

A. I only became --

MR. KWON: Objection, Your Honor. The question lacks foundation.

THE COURT: She is asking if he was aware of it.

While you were at Sun, were you aware of anything like that?

THE WITNESS: No.

THE COURT: All right. Okay. So he has no personal knowledge of that.

BY MS. SIMPSON:

Q. And you testified about Apache earlier today, too; correct?

A. I did.

Q. Apache pursued its Harmony project pursuant to a specification license with Sun, did it not?

A. It would be very difficult for me to answer that question.

MS. SIMPSON: I'd like to show the witness trial Exhibit 9191. This has already been admitted.

THE COURT: What part do you want him to look at?

BY MS. SIMPSON

Q. This is an email that you received between you and Jonathan Schwartz; correct?

A. It is.

Q. And if you turn to the second page of the document, it's actually an email that is being sent from Geir Magnusson to Jonathan Schwartz. You testified earlier that Geir Magnusson was the officer at Apache Software Foundation; correct?

A. That -- his role was to represent Apache Java in relation to Java matters, yes.

Q. And if you look on the back of the second page of that document, at the very bottom. Do you see where I am? (reading)

"Through Apache Harmony, the ASF," that's the Apache Software Foundation, "entered into the specification license in good faith with the expectation that Sun, as the spec lead, would reciprocate."

Do you see that?

A. I do see that.

Q. So Geir Magnusson is indicating that Apache took a specification license, is he not?

THE COURT: Well, wait. You said that so fast. Say that again but more slowly.

MS. SIMPSON: Sorry.

Q. Geir Magnusson here is indicating that Apache had a specification license for the Harmony project, is he not?

A. That is what it appears to say.

Q. And I'd also like to look at a document Trial Exhibit 2207. It should be up in your folder, Mr. Phipps.

A. Okay. Yes, indeed.

Q. Did you find that?

A. I did.

Q. Can you take a look at it, sir? Is that an email chain between you and Jonathan Schwartz?

A. It is, yes.

Q. And it also contains email from Geir Magnusson?

A. (Witness examines document.) Yes. Yes, it is.

MS. SIMPSON: Move to admit this into evidence, Your Honor.

THE COURT: Any objection?

MR. KWON: No objection.

THE COURT: All right. Thank you. It's in, but I don't have the number.

MS. SIMPSON: 2207.

THE COURT: 2207 in evidence.

(Trial Exhibit 2207 received in evidence)

BY MS. SIMPSON:

Q. And this is dated April 10th, 2007; correct?

A. It is.

Q. And if you turn to the second page of the document, Mr. Phipps, where it begins with Geir Magnusson writing to Jonathan Schwartz, do you see the second paragraph that begins "Since August 2006"?

A. Yes.

Q. It says: (reading)

"The ASF," the Apache Software Foundation, "has been attempting to secure an acceptable license from Sun for the test kit for Java SE. This test kit called the Java Compatibility Kit or JCK" --

Folks also refer to that as the TCK; right, Mr. Phipps?

A. I believe they do, yes.

Q. (reading)

-- "is needed by the Apache Harmony project to demonstrate its compatibility with the Java SE specification as required by Sun's specification license."

Do you see that?

A. I do.

Q. Did you understand that the Apache Software Foundation needed a TCK in order to implement the specification?

MR. KWON: Objection. Calls for a legal conclusion.

THE COURT: Well, if back then you had a view on that

subject, you may answer the question, notwithstanding that it calls for a legal conclusion, but you shouldn't be spinning off some new theory now. It's what you actually thought about and understood back at the time that would count, unless you got that from a lawyer, which would mean it would be privileged and you shouldn't be talking about it.

So can you answer that question based on your historical memory?

THE WITNESS: I probably can.

THE COURT: All right. Go ahead and try.

THE WITNESS: So I don't recall believing that Apache needed to have a specific license to implement Harmony.

BY MS. SIMPSON:

Q. You don't remember a dispute in the industry, a public dispute, about Apache wanting a TCK and Sun refusing that license?

A. Oh, absolutely I remember that, but that was after they implemented it. They didn't need it in order to make the implementation.

MS. SIMPSON: Your Honor, I'd like to approach the witness with Exhibit 9193.

THE COURT: Sure.

THE WITNESS: Thank you.

BY MS. SIMPSON:

Q. This is an email from Bill Shannon to Ray Gans dated

December 15th, 2005, and you see that you're cc'd on this document, Mr. Phipps?

A. Yes.

MR. KWON: Your Honor, we'd object to this document. It's an email that indicates it has no attachments, but then there's something else attached to it. It's not a document as it existed in the ordinary course.

MS. SIMPSON: Your Honor, the attachment is listed as a link in the document and the linked document is attached to the email.

MR. KWON: Your Honor, that's incorrect. There's a link in the email -- first of all, a link in the email and a separate document, it's still two documents. But, second of all, the URL on the document that accompanies it is not the same as any of the URLs in the email.

THE COURT: Well, was this sent to the witness?

MS. SIMPSON: Yes, Your Honor. It's a cc. He was a cc.

THE COURT: So the extra pages here is something you stapled on?

MS. SIMPSON: It's the link, Your Honor, where it says -- do you see how there's "See the following for more information"?

THE COURT: Well, these ought to be separate documents. I'm going to let you use 9193, but I'm going to

pull off the attachment, and then you can try to get the attachment in in some other way, but that's not really the same document. So if you can establish that he read this attachment, then you can get that in, too.

We can call them 9193A and B. So let's stick with A for the moment.

(Trial Exhibit 9193B marked for identification)

BY MS. SIMPSON:

Q. Mr. Phipps, did you see the reference to "See the following for more information" in the middle of that document?

A. I do.

Q. And do you see those links there?

A. I do.

Q. Did you read those links when they were sent to you?

A. I have no recollection.

Q. Okay.

THE COURT: We'll let 9193 come in evidence, but so far B is not in evidence.

(Trial Exhibit 9193 received in evidence)

THE COURT: All right.

BY MS. SIMPSON:

Q. You testified, Mr. Phipps, that open source could be used without permission and without restrictions. Did you say that this morning?

A. I made a general statement along those lines, yes.

Q. But open source always requires a license, does it not? Every open source community has a license attached to it; isn't that correct?

A. So I wouldn't want to pretend to be a legal expert, but open source communities do prefer a copyright license under which they like their work to be licensed, yes.

Q. And those licenses have certain restrictions attached to them; correct?

A. So those licenses have a set of terms under which they grant a copyright. I think the use of the word "restriction" is quite controversial in the open source community, so I'd be very cautious about using it.

But, yes, there are licenses that have license terms that give those four freedoms -- the freedom to use, to modify, to share, and to share the modified form -- and they do that in a whole range of different ways. There's about 70 different open source licenses that do it in different ways with more and less verbiage around them.

Q. And speaking specifically about Sun and OpenJDK, Sun released open JDk pursuant to a license; correct?

A. OpenJDK was released under the GPLV2 yes.

Q. And that GPLV2 which released OpenJDK, it also licensed the declaring code and the structure sequence and organization of the 37 packages at issue in this case; correct?

A. Yes.

MR. KWON: Objection, Your Honor. Calls for a legal conclusion.

THE COURT: He's already answered. I think he already said that it was part of OpenJDK; right? Right?

MR. KWON: It's a different question, Your Honor, what is part of OpenJDK and what is licensed by the license.

THE COURT: Well, I'm going to allow the answer to stand.

I have a question on this, but I don't want to interrupt your flow, Ms. Simpson. Would you go ahead and I'll ask it, but it does bear directly on this point.

BY MS. SIMPSON:

Q. Mr. Phipps, Sun had a number of other licenses through which it licensed the declaring code and the SSO -- which we've shortened from structure, sequence, and organization -- at the time -- and I'm talking about the 2005 to 2007 time frame -- correct?

MR. KWON: Your Honor, same objection. He hasn't testified about this before.

THE COURT: Well, it doesn't matter if he's able to testify about it. It's cross-examination.

Go ahead and answer.

THE WITNESS: I'm aware there were other licenses.

BY MS. SIMPSON:

Q. So Sun had a commercial license in the 2005 to 2007 time

frame where it offered the declaring code and the SSO at issue in this case; isn't that correct?

A. That's a matter that I don't have direct understanding and knowledge of. It's generally known that there were other licenses, but I focused on open source.

Q. But you're aware that Sun had commercial licenses?

A. I'm aware that Sun had commercial licenses.

Q. Indeed, we just looked at the FAQ where we talked about how Sun was going to continue offering those commercial licenses?

A. But I wouldn't want you to think I was an expert on them.

Q. And Sun also offered the declaring code and the structure, sequence, and organization pursuant to the specification license that we've been talking about; correct?

A. I don't know that I can answer that question.

MS. SIMPSON: Your Honor, permission to read into the transcript -- or into the record from the deposition of Mr. Phipps taken May 7th, 2016, page 45, lines 19 to 25.

THE COURT: Any objection?

MR. KWON: Yes, Your Honor, I object. He's a third-party witness, and it's not true impeachment.

THE COURT: Well, he's being paid by -- well, maybe not today, but in the past paid by Google, so that's close enough for my purposes. I'm going to allow it to be read.

Go ahead, read it, Ms. Simpson.

MS. SIMPSON: (reading)

"Q. Right. So during the time period that you were at Sun, Sun made the Java APIs, including the declaring code and SSO, available for use by others pursuant to the specification license; true?

"A. Under this -- under this specification license and other arrangements."

Q. In fact, Mr. Phipps, while you were at Sun, Sun made the declaring code and the SSO of the Java APIs available only through a license; isn't that correct?

THE COURT: Well, that is ambiguous because are you -- the words "a license," does that mean -- are you including the GPL Version 2 with Classpath exception, or -- it's unclear what you mean there.

BY MS. SIMPSON:

Q. You needed a license of any kind to use the SSO and the Java APIs -- of the Java APIs and the declaring code; isn't that correct?

MR. KWON: Objection, Your Honor. This is legal testimony.

THE COURT: Well, are you -- sustained. It is legal testimony because you're leaving out the "fair use" part. If you want to put that part into your question, I guess I would allow it, but otherwise, as a legal question, it's not complete.

MS. SIMPSON: Your Honor -- I'm sorry.

Q. Mr. Phipps, can you cite any entities from the 2005 to 2007 time frame whom used the Java APIs in a commercial product without taking a license from Sun? You can't do that, can you?

MR. KWON: Objection, Your Honor. Goes beyond the scope of the direct examination.

THE COURT: Well, but I can -- if I sustain that, then I'd have to have him come back for their case. I'm going to overrule the objection so that he can just make one appearance.

Please answer.

THE WITNESS: If you could repeat the question again.

BY MS. SIMPSON:

Q. You cannot identify any entity in the 2005 to 2007 time frame who used the Java APIs in a commercial product without taking a license from Sun, can you?

A. I believe that if I had sufficient notice of that question and could do some research, it may be possible to identify them; but in the court today, no, I can't.

Q. And you haven't thought to do that despite the fact that you've been retained by Google in this matter on these very issues?

MR. KWON: Objection, Your Honor. Misstates the nature of the consultation.

THE COURT: All right. Sustained. Sustained. But you could rephrase that.

BY MS. SIMPSON:

Q. What is the nature of your consultation agreement with Google, Mr. Phipps? What did they retain you to do?

A. They retained me to advise them on what was going on in the open source team -- in the open source community, rather, around OpenJDK, and to help them find documents that Oracle had deleted from the website related to that, including the -- the FAQ.

Q. And they flew you here to testify today; correct?

A. Sorry?

Q. They flew you --

A. Oh, they flew me here, and I've been waiting for quite a long time in a hotel to come and talk to you.

THE COURT: Well, I hope you've had a good time in San Francisco. Most people do.

THE WITNESS: I do like it here.

THE COURT: All right. Good.

BY MS. SIMPSON:

Q. Mr. Phipps, even though Google retained you in this case, upon your departure from Sun in 2010, the Google employees had very negative things to say about you, didn't they?

A. Oh, I haven't --

MR. KWON: Objection, Your Honor. Foundation. 403.

THE COURT: Do you know?

THE WITNESS: No idea.

THE COURT: All right. Then, now, this is a good example. The lawyer has said it. Is that evidence? No, because the witness says he doesn't know. Maybe that could be proven up in some other way; but just because the lawyer says there were negative comments, so far that's not evidence. Please remember, that's a good example.

Okay. Ms. Simpson, continue on.

MS. SIMPSON: I would like to approach the witness, Your Honor, with Trial Exhibit 9190.

THE COURT: All right.

THE WITNESS: Thank you.

MR. KWON: Your Honor, I'd object to the use of this exhibit at all on the same foundation and 403 grounds and hearsay.

THE COURT: Is this going to be internal Google documents with negative comments about the witness?

MS. SIMPSON: Your Honor, this is a document we looked at the other day.

THE COURT: I don't remember the document we looked at the other day, but let me see where we are.

Any part of this that he saw himself, you can use; but just because other people in Google made comments that he didn't know about, or unless he did know about them, that's not an occasion to lay that before the jury.

MS. SIMPSON: Your Honor, this goes directly to

Mr. Phipps' testimony.

MR. KWON: Your Honor, there's no foundation for any of the parts that she wants to actually use.

THE COURT: I sustain the objection. You can use any part of that document he actually saw and received at the time. That would be okay.

But the other comments are extraneous to this witness. Even though they may relate to him, he never saw them and has no firsthand knowledge of it, I assume. You can try to establish that, if you can, but be careful because it's a document not in evidence.

MS. SIMPSON: Well, Your Honor, Google has held him out as an expert in advising them on these areas, and this goes to his knowledge.

THE COURT: Well, it's still not a basis to get around the foundational rules of using this document.

MS. SIMPSON: All right, Your Honor. I pass the witness.

THE COURT: Thank you.

Before you -- I have a question for the witness. And you said -- you drew a distinction earlier between an API and a library.

THE WITNESS: Yes.

THE COURT: Did I understand you correctly?

THE WITNESS: Yes.

THE COURT: All right. Say again what the API in your view is versus library.

THE WITNESS: Right. Well, so the class library is the implementation of the code. So it's -- that's where all the know-how that makes the thing that is desired to happen happen.

THE COURT: Then what is the API?

THE WITNESS: The API is the line of code that allows somebody outside to gain access to that function.

THE COURT: Is that what we're been calling the declaring line of code?

THE WITNESS: So in the program -- when you're writing a program, that line of code is the declaration, and then that same line of code appears above the source code of the library as well. So that appears in both places, but it gets used independently of the implementation when you're actually writing a program.

THE COURT: So when you refer, anyway, to API, are you referring to that collection of declaring lines of code?

THE WITNESS: Yes. So I'm referring to, for example, a function prototype, the name of the function, the list of parameters that it calls.

THE COURT: Is that also known as a specification?

THE WITNESS: Well, the specification -- so in Java in particular the specification tends to be in a block of comments

underneath the function declaration, and then there's a tool that automatically extracts those comments to produce the specification.

So the -- you have -- the API is the function call that gives you access to the library. The specification tells you what would happen if you were to make -- if you were to call that function.

THE COURT: All right.

THE WITNESS: And then the library is the thing that actually does it when you call it.

THE COURT: Okay. Now, with that distinction in mind -- and I'm not saying that that is the way other witnesses have or have not used it, but that's the way you're using it, that's just preliminary to the question -- you said that you open sourced -- while you were at Sun, you open sourced Java on with something called OpenJDK.

THE WITNESS: Yes.

THE COURT: All right. And that was under the GPL Version 2 with Classpath exception; is that true?

THE WITNESS: Yes.

THE COURT: All right. Now, when you open sourced it, was that the API part meaning just the declaration lines or did that also include the libraries?

THE WITNESS: No. That included the full source code to the libraries. So that was everything -- the source code to

the library contains the declaration line, the comments that form the specification, and then all of the implementation, and what we released under the GPL was all of that. And under the GPL, obviously, you can use any part of it that you want to because there's no restriction on breaking it up.

THE COURT: All right. Okay. Ms. Simpson, I'm not charging this time against you, but if I have asked a question that you want to follow up on, go right ahead now, and then I'm going to -- then counsel can do the same thing.

MS. SIMPSON: I'm all right. Thank you, Your Honor.

THE COURT: Good.

Okay. Mr. Kwun, your turn.

REDIRECT EXAMINATION

BY MR. KWON:

Q. If you could take a look at Trial Exhibit 2007.

A. 2007.

Q. Excuse me. 2207.

A. Two two --

Q. It's one of the exhibits you've already looked at.

A. Right. Quite a lot of them here.

(Witness examines document.) 2207, yes.

Q. If we could get that up on the screen. It's in evidence. Can we get page 2 of the exhibit on the screen?

And can you look at page 2, Mr. Phipps?

A. Yes.

Q. You were looking at a portion of this email from Geir Magnusson to Jonathan Schwartz, and there's a paragraph that says: (reading)

"Since August 2006, the Apache Software Foundation has been attempting to secure an acceptable license from Sun for a test kit for Java SE."

And it refers to it as the JCK. Is that sometimes also referred to as a TCK?

A. Yes.

Q. If Apache had gotten a TCK or JCK license, what would that have allowed it to do?

A. The main practical use would have been that they would have been able to apply to use the Java-compatible branding, so they would have been able to put the steaming coffee cup onto their implementation of Java SE.

Q. If you could turn to the last page of this exhibit, page 4.

A. Okay.

Q. There's a paragraph at the bottom or toward the bottom that says: (reading)

"Through Apache Harmony, the ASF is implementing Java SE in good faith."

And if you could also take a look -- keep this one open, but if you could also take a look at Exhibit 9191.

A. Okay.

Q. You were shown a paragraph at the end of that one on page 2 of 9191 that says: (reading)

"Through Apache Harmony, the ASF entered into the specification license in good faith."

Do you see that?

A. I do.

Q. What is the difference in that -- just in the beginning of that, what has changed between those two emails?

A. Well, at some point between these two emails -- let me see, which is the -- one of them is on February the 9th and one of them is on April the 10th, and the later version Geir doesn't mention the specification license in the text of the email. So that later one I think reflects the -- the public letter that he posted on the website about that time.

Q. And if we could look at -- if you could grab 7722 and turn to page 9. Ms. Simpson asked you some questions about this document on page 9. And there's a question there that says "Do you think anyone will fork the JDK?"

A. I see it.

Q. Was there anything in the license terms that prevented people from creating incompatible forks of OpenJDK?

A. No.

Q. If you could look at the next question below that, the one with the four bullet points.

A. Okay.

Q. The question is, "So what about compatibility?" And there's the four bullet points, and then beneath that it says: (reading)

"In addition to the specific steps that we're taking to foster compatibility, we are convinced that the market will demand compatible implementations and that incompatible ones won't gain traction."

Was it Sun's strategy to foster compatibility rather than require compatibility?

A. Absolutely, yes.

Q. And you were relying on the market to help you out in that respect?

A. Yes. That was the compromise we had to make to use an open source license on Java.

MR. KWON: I pass the witness.

THE COURT: Ms. Simpson, anything more.

MS. SIMPSON: No further questions, Your Honor.

THE COURT: May this witness be excused?

MR. KWON: Yes, Your Honor.

MS. SIMPSON: Yes, Your Honor.

MR. VAN NEST: May he be released from subpoena as well, Your Honor?

THE COURT: That's what I meant.

MS. SIMPSON: Yes.

THE COURT: Okay. You're free to go. You can go back

to England.

THE WITNESS: Jolly good. Thank you very much, lovely though it is here.

THE COURT: All right. Have a great day.

(Witness excused.)

THE COURT: Okay. We're going to take our 15-minute break in just a moment.

So just are we going to have more live witnesses after the break?

MR. VAN NEST: Both.

THE COURT: Both live and video. Okay. Great.

So see you in 15 minutes.

(Proceedings were heard out of presence of the jury:)

THE COURT: All right. Have a seat.

Let me just ask you-all who -- both sides who know, was he correct that both the implementing code and the declaring code were open sourced and under OpenJDK?

MR. VAN NEST: Yes.

MR. KWON: Yes, Your Honor.

THE COURT: All right. I'll just -- I still, after all this time, am a little mystified why anyone would want to re-implement anything. If you could get it under that license, why would you go to the trouble to re-implement anything then? Why not just use it the way it's in OpenJDK?

I'm just thinking out loud. I commend you that question

because I bet you members of the jury are wondering what's going on there. Why wouldn't somebody just use OpenJDK as is? There must be some reason.

MS. HURST: I think Mr. Rubin answered that question yesterday, Your Honor.

THE COURT: What did he say?

MS. HURST: He said that it would not have been acceptable to the handset manufacturers to use that license.

MR. KWON: Your Honor, that's not what Mr. Rubin said.

MS. ANDERSON: Actually, Your Honor, Mr. Rubin testified at the stand that what happened is OpenJDK was released in about May of 2007. Android came out in November of 2007. By May of 2007, Google had been working on their implementation for years. It was too late to do anything about it. There was no need to do that.

So, you know, this is an after-the-fact situation where years later Oracle's claiming that declarations are not freely usable, which was contrary to the entire practice. There was absolutely no reason to do anything with OpenJDK under those circumstances when we already had our implementations in place and everyone understood the declarations were free for use.

So at the time, you know, the fact that Sun came out in May of '07 with their own release of OpenJDK, great, but we were well down the road. You have to kind of change things if you're going to plop in new code. So that's the state of play.

MS. HURST: Well, we're going to see more about that with the next witness, Your Honor. I'll get you the transcript. I know the witness said that the GPL was unacceptable to the handset manufacturers yesterday, and we'll get you the transcript cites for that.

MS. ANDERSON: Thank you, Your Honor.

THE COURT: Well, who is the next witness?

MS. ANDERSON: Our next witness is Dan Bornstein, Your Honor. So he was one of the folks working on the code at Android, at Google at the time.

THE COURT: All right. Do you have the breakdown on the deposition read-in?

MR. VAN NEST: Yes, I do, Your Honor.

THE COURT: All right. What is it?

MR. VAN NEST: After a lengthy negotiation, I have the results.

THE COURT: All right.

MR. VAN NEST: On Mr. Duimovich, it's 50/50.

THE COURT: Wait a minute. That's 17 minutes. I'm going to charge each of you 8 -- I'll just make it 8 apiece. Okay.

MR. VAN NEST: Are you sure? Is that Duimovich or Smith you're on there? 17 minutes?

THE COURT: It was 17. I keep track of it up here.

MR. VAN NEST: Okay.

THE COURT: Go ahead.

MR. VAN NEST: The parties think it's 12, but okay. But its 50/50.

THE COURT: I mean, there's dead time in between. Somebody's got to absorb that.

MR. VAN NEST: All right. Then on Smith it's 70 percent Google, 30 percent Oracle.

THE COURT: All right. So that one was 8 minutes. 8 times 70 is 5.6. So you get 70 percent?

MR. VAN NEST: Yes.

THE COURT: All right. You get 6 and they get 2. See how easy it is to resolve these things?

All right. And then on --

MR. VAN NEST: On Mr. Ellison, that's a hundred percent.

THE COURT: All right. I've got everything now under control. Great. Thank you.

(Recess taken at 11:24 a.m.)

(Proceedings resumed at 11:38 a.m.)

(Proceedings were heard out of presence of the jury:)

THE COURT: Are we ready to go?

MS. ANDERSON: Yes, Your Honor.

THE COURT: Please get the witness and please bring in the jury.

(Proceedings were heard in the presence of the jury:)

THE COURT: Google may call its next witness.

MS. ANDERSON: Thank you, Your Honor.

Google calls Mr. Bornstein, who is already at the stand.

THE COURT: And does he have a first name?

MS. ANDERSON: He certainly does. Mr. Daniel Bornstein.

THE COURT: All right. Please, Mr. Bornstein, welcome, and raise your hand, please.

DANIEL BORNSTEIN, DEFENDANT WITNESS, SWORN

THE CLERK: Please state your name for the court and spell your last name for the record.

THE WITNESS: Daniel Bornstein.

THE COURT: Your hand can go down.

THE WITNESS: Oh, sorry.

THE COURT: All right. Now, you've said your name. And you need to be about this close to the microphone. That's going to work.

Okay. Go ahead, Counsel.

MS. ANDERSON: Thank you, Your Honor.

DIRECT EXAMINATION

BY MS. ANDERSON:

Q. Good morning, Mr. Bornstein.

A. Good morning.

Q. Would you please introduce yourself to the jury and let them know who you are generally in relation to this case?

A. I'm Dan Bornstein. I'm a computer programmer. I've been programming most of my life, and I worked at Google on the Android project from 2005 through 2011.

Q. Thank you.

Where do you live today?

A. I live in San Francisco.

Q. And how long have you been in Northern California?

A. Since 1992.

Q. You mentioned you worked in the computer industry. How long have you been involved in computers?

A. About since I was like seven years old.

Q. All right. Have you written any computer programs during your life?

A. Many.

Q. When did you write your first program?

A. Probably around seven years old or so.

Q. All right. About how many times in your life do you think you've written a computer program?

A. Thousands. Yeah. Computer programmers just kind of write programs left and right.

Q. Okay. Would you tell the jury how many computer programming languages you know how to program in?

A. Also a lot. A dozen or so, maybe more.

Q. Would you give the jury some examples, please?

A. Sure. Let's see... Well, certainly Java, C, C++, Basic,

Pascal, Shell Scripting, AWK. Let's see, a bench of assembly languages, like 6502, 68K, ARM, Spark.

Q. Thank you.

A. I'm sure I've missed -- Logo -- one or two more.

Q. Thank you very much.

Do you have a degree from any university?

A. I do.

Q. What is your degree and where did you earn it?

A. I have a Bachelor of Science in cognitive science from Brown University.

Q. Would you please tell the jury where you've worked since you graduated from Brown?

A. Many -- many places. I've, yeah, again, had, like, maybe 10, 12 jobs since graduating.

Q. All right. And could you give some examples of the places you've worked over the years?

A. Sure. Almost all of them have been software companies of one form or another. I started my career at a company called Kaleida Labs, which was a joint venture between Apple and IBM.

I worked at a company called Electric Communities doing virtual reality-ish stuff.

A few jobs later I worked at a company called Danger that was doing an early smartphone. After that was when I worked at Google.

And since then, I've worked at a couple other companies.

One is now called Medium, and I now work for a company called Slack.

Q. Thank you.

Would you please tell us, what years did you work at Danger?

A. Danger I worked -- I worked there from 2002 to 2005.

Q. You said you worked there on an early smartphone. What did you mean by that?

A. I mean at the time -- I don't even think "smartphone" was a term back then. Now it's pretty commonly understood. I think of the work that I and my colleagues did at Danger as really the kind of predecessor to the current smartphone revolution.

Q. And during what years -- would you remind us again -- what years were you at the company Google? When did you work there?

A. Oh, Google I was at from 2005 through 2011.

Q. What was your title during the time you worked at Google?

A. I had a few titles. I think my original title was senior software engineer. I eventually got the title staff software engineer. I was also a tech lead at -- during part of my time there -- most of my time there. I was also a people manager during part of my time there.

Q. And since the time that you left Google in 2011, have you since been retained by Google to do any consulting work in advance of this trial in relation to this case?

A. I have.
Q. All right. When did you learn the Java programming language?
A. So I think I learned of the Java programming language at that first job out of college, probably around '94 or so, and I think I actually learned -- well, I had to use it for my job for the first time in 1996 at Electric Communities.
Q. And what did you use the Java language to do while at Electric Communities?
A. Electric Communities, as I said, was building a virtual reality kind of platform. The idea of it was that people -- developers could develop new bits of interactive stuff in this system, and Java was being used as the platform there for a lot of that stuff.
Q. How did you learn how to program in the Java language?
A. I learned about the -- the way a programmer learns any new programming language, which is a combination of things. You know, reading books, reading documentation of various forms, seeing -- seeing other people source code.
Q. Okay. And for the learning the Java language, what material did you read to learn that language?
A. Well, there are a couple books in particular that I read. One was an O'Reilly book called *Java in a Nutshell*, and I also read the *Java Language Specification*.
Q. Before you, Mr. Bornstein, are a stack of a few documents

with trial exhibit numbers on them. Would you please take a look at Exhibit 4027?

A. Uh-huh.

Q. That's it right there.

A. Yeah.

Q. Do you recognize this exhibit?

A. I do. That looks like the cover page of the book, the *Java Language Specification*, the one I was just mentioning.

Q. Great.

MS. ANDERSON: And this exhibit is already in evidence, Your Honor, so we'd like to put it up on the screen.

THE COURT: All right. Go ahead.

MS. ANDERSON: Thank you.

Q. Have you read Exhibit 4027?

A. I have.

Q. Okay. Literally have you read the entire book?

A. Literally I've read the entire thing.

Q. And when you were reviewing this material, what kind of information, just at a high level, did you learn from reviewing a book like Exhibit 4027?

A. So what I learned was, in part, how to program in the Java programming language.

Q. Okay. Did you use your knowledge of the Java programming language that you learned from books like this to do your work at Google on the Android platform?

A. Yeah. I hadn't read this in quite a while by the time I was at Google, but certainly my knowledge of it carried forward.

Q. Okay. In the course of your learning the Java programming language, did you have reason to learn what Java APIs are?

A. Yeah.

Q. At a high level, would you identify what your understanding was of what a Java API is?

A. Okay. So first of all, the term "API" isn't just -- isn't specific to Java, so the Java APIs are the APIs for the Java programming language. And when the API is -- it's kind of the set of stuff that you can use as an application programmer that you can kind of expect the system that you're running to provide for you.

Q. Have you heard words like "methods," "classes," and "packages" used in connection with the idea Java APIs?

A. I have.

Q. And when you heard these things, what did you learn or understand them to be?

A. What I understood them to be is the way the Java programming language is set up, it partitions the API into different subdivisions. Each of those subdivisions gets a name, and the name is a form a hierarchy of sorts.

So at the top level is a package name, which itself can sort of have partitions; and within a package there are a

number of classes. Each of those classes also has a name. And then when you sort of drill into the class, you find that there's also a bunch of other bits and pieces that the class makes up or that make up the class, and methods are among the things that make up the class.

Q. Okay. And in your exposure to and work with the Java programming language, when you hear the word "Java API," does it only mean the reference to a single method? Is that the only way it's used?

A. No. An API, it depends on context. An API can refer to just a little piece of an implementation or piece of -- or sort of a piece of the tool kit. It can -- and it can sort of, like, grow its way up to refer to a whole class or a whole package or even a whole set of packages.

Q. And in the course of learning this language, did you come to learn what declarations are for Java APIs?

A. I did.

Q. And would you describe what that is to the jury?

A. So just as before, a declaration is something that's independent of Java in particular, so Java declarations are the declarations that one would find within a set of Java code. And, say, with respect to an API, the declarations are sort of -- give, like, the names -- the names of these things and sort of like the patterns of how you have to use them in order for application code to successfully talk to a library

implementation.

Q. And how about implementing code? Does that have anything to do with the subject of APIs and declarations?

A. Well, implementing code is not the API declaration. It's a separate piece. It's related. The implementing code for a particular piece of an API is generally kind of tied -- you know, it's usually -- usually, let's say, it appears in a file nearby the declaration, and it's meant to be a reflection in some sense of the -- of the declaration, but it's not the same thing.

MS. ANDERSON: Okay. With the Court's permission, Your Honor, we'd like to ask if the witness could demonstrate some of his testimony on the easel. May he approach if we move it forward?

THE COURT: All right.

MS. ANDERSON: Thank you, Your Honor.

THE COURT: I'm just curious. Is he designated as an expert witness?

MS. ANDERSON: Well, he actually is, Your Honor, although he can also demonstrate from his knowledge at the time that he was working on the team.

THE COURT: Well, I know, but which role are we in here?

MS. HURST: He was not designated as an expert, Your Honor.

MS. ANDERSON: He is, Your Honor. He is an employee expert. He was disclosed in a document that I have here; but, nonetheless --

MS. HURST: That was at the time of a prior proceeding, Your Honor.

MS. ANDERSON: Your Honor --

THE COURT: Well, then just stick with what he knew back at the time he was doing his job --

MS. ANDERSON: I will do that.

THE COURT: -- and not try to get into present-day opinions and explanations.

MS. ANDERSON: I absolutely will, Your Honor.

THE COURT: All right. So he has testified that he knew the language and all of this information back at the time he was working at Google, so -- but you've got to stick to that time frame.

MS. ANDERSON: We will.

Q. All right. Mr. Bornstein, if you could approach this easel we have here, and I'd like to draw your attention to the time you were working with Google and on the Android team using the Java programming language. Were you familiar with examples of Java APIs at the time?

A. I was.

Q. Could you please demonstrate for the jury your knowledge and explaining what the terms you've been using mean in

connection with an actual API you used and knew about when you were with the Google team on Android?

A. Sure. I'm going to use a class called math, which is about mathematical operations and a method called max, which is short for maximum. The idea of the max API is that you can hand it two numbers, and it hands you back whichever one is larger. Usually you'll hand it a variable so you don't -- when you're running the total, so you don't know what the -- like, what the actual values are. So it's sort of, like, you hand it two numbers, you know you'll get back a number, and you know it will be the bigger -- that it will be the bigger of the two numbers.

So I apologize for my bad art skills. It kind of stopped when I was in the third grade.

This is me at my computer. And I'm thinking, I've got a problem, and one way to partially solve that problem is to figure out which of two numbers -- of two numbers is bigger.

So I'm thinking, okay, I've want to solve that problem. I have X and Y. Like, I don't know what the actual numbers are. I'll just call them X and Y. So typing at my computer. Because I know -- at the time I knew the Java programming language, I knew that this thing called math.max existed.

And, you know, why did I know it? You know, I read the book, saw Web pages, whatever. So somewhere in there was the thing that said math and somewhere in there it said max, and

there's a lot of paragraphs of text that said, like, "This is how you use it," that sort of thing.

And so then I got to typing, and so I got -- I had -- somewhere in my code I had some X and Y already defined, and I got to say something like int M equals math.max of X, Y.

And so what this would cause is the variable M would get whichever one of these values was the bigger one. That would be how I would, as an application programmer, use math.max.

Q. Okay. And could you, first of all, maybe in a different color, write on this what the name of the method you were writing about?

A. Oh, yeah. So the -- so the class that the method is in is math and the name of the method within that class is max.

Q. Okay.

A. This is the class name (indicating). This is the method name (indicating).

And we also talked about packages. The package -- in this case, the package is kind of implied, but I happen to know as a Java programmer that the package is java.lang.

Q. Now, so far, is there anything you've written on here so far that would be considered an API depending on context and back at the time?

A. Yeah. So what I would say is that the API is sort of like this thing in my head about, like, how do I get these -- get this action to take place, and what I've typed is an

application use of the API.

Q. Okay. And what's the name of the API?

A. So in this case, I would call it, say, just, like, math.max.

Q. And have you drawn anything yet that would be considered declarations or declaring code?

A. No, I haven't.

Q. All right. So could you now draw for us something that you understood at the time would be declarations or declaring code for this particular API max?

A. Sure. So this would be in a different file somewhere. So it would be in the library implementation. And so I'm going to try to give you as accurate and complete a picture as I can in this space here.

So at the top of a file in Java is a package declaration. Package, like I said, it was -- it's java.lang. And it's going to name the class. So class. So it's saying this is the class.

"Public" means it's being exposed to user -- to application users or application developers. Public class on that. So this is a curly brace, which is used a lot in Java. I'll try to make it legible.

And there is going to be a whole bunch of stuff in math that has nothing to do with max, but at some point there will be in here public static max, and it will define in parentheses

something like int A, comma, int B, and another curly brace; and then there will be the implementation of it, which you didn't ask about so I won't write it, but --

Q. Save yourself enough room for the implementation.

A. I thought you might -- so then -- so there's going to be something here that implements it, and then there is going to be more stuff that has nothing to do with max, and then there's going to be a final closing brace to close off the class. So that's, more or less, what it would look like.

Q. And could you please fill in on your drawing here what you would have known to be a possible implementation of max at the time you were with the Android team?

A. Sure. I'll give you pretty much the most straightforward possible implementation in Java. So if A is greater than B, return A, else, return B. There you go.

Q. Great.

All right. So now if you could take another color, just so it's easier to see, and could you identify for us with some kind of bracketing where a declaration would be on what you've drawn, and back again at the time you were with Android?

A. Yeah. So this part is the package declaration (indicating), so this is named the package. This is the class declaration (indicating). So here's naming of the class. And here is the method declaration (indicating).

Q. Okay. And then maybe could you point to it indicating

declaration and --

A. (Witness complying.)

Q. Perfect.

And now could you take a different color and identify for us where the implementing code or implementation would have been in this example from your time at Android?

A. Yep. So this right here (indicating) is the implementation. Too many letters. Okay.

Q. And if you can, maybe step either a little more that way or to this side, whatever side is easier just so the whole jury can see.

There you go, so everybody can see.

With respect to what you've identified as the declaration for the particular method max, that's the one that starts public static; is that what you said?

A. Yeah. That's right.

Q. When you were with Android, were there rules that governed how that had to be written?

A. Yeah. Absolutely.

Q. And could you give us an explanation of what kind of rules you were operating under at the time?

A. Oh, sure. Okay. Math.max is something that is available for application programmers to use, which means it has to be declared public. So this word "public" is required.

"Static" means that -- it's something that we haven't

really talked about here, but there is two different sort of categories of methods in Java. It so happens max has to be a static one given the way it's defined for programmers. So this word has to be there.

The name is -- the name is the name, so there's no choice in what the name is.

Parens also have to surround these things called parameters.

The types of the parameters are well-defined. So as I said, int is short for integer, a kind of number. And then this names the parameters.

Q. Thank you.

And in terms of flexibility, how would you have characterized the rules governing how one writes declarations back at the time at Android?

MS. HURST: I'm going to object to that question, Your Honor. I don't know what "rules" means in this context.

THE COURT: It's also present-day opinion. Sustained. You can rephrase it.

BY MS. ANDERSON:

Q. Back at the time you were with Android when you were working on the Java language, what did you understand to be the amount of flexibility you had in how you would write a declaration for a method like max?

MS. HURST: Objection. Leading.

THE COURT: No. It's not leading. That's overruled. Please answer.

THE WITNESS: I would say that there's very little flexibility.

BY MS. ANDERSON:

Q. Why is that?

A. So the names of the -- the places where these names occur, there's -- there's fixed positions -- there's some flexibility, like these two could have been reversed. There's a standard way of doing it. So it could have been, but probably wasn't, the name just -- the name had to go here. There's no other choice about that. The parens had to go there. No other choice about that.

These names of these types, the int, there was no choice about that. And there was, say, some flexibility in the names of these parameters. That would -- I think that covers the flexibility.

Q. And, again, back at the time you were with Android, what did you understand to be the relative amount of flexibility available to you in writing the actual implementation for this method?

A. For the implementation?

Q. Yes?

A. Much more flexibility. That's where more of, say, a programmer's experience and taste in sort of design limitation

would come into play.

Q. We've heard the phrase "independent implementation." How, if at all, does that relate to the testimony you just gave, back at the time you were with Android?

A. Oh, at the time. So an independent implementation, it's talking about what would go here and, of course, there's lots of packages, lots of classes. So it's sort of all of the peers for the many, many methods in -- in the -- that are defined by the API. So an independent implementation is just somebody taking the knowledge that they have about how Java works and writing a whole bunch of new code to implement it.

Q. When you said "here," you were pointing to which box?

A. The orange implementation box.

Q. All right. Thank you very much.

So I think you can take the stand again. Thank you, Mr. Bornstein.

Putting you back again to the time you were on the Android team, at the time did you have a view as to whether or not you were free to use the Java programming language?

A. I did.

Q. What was your view?

A. I believed I was free to use the Java programming language.

Q. Why did you believe that at the time?

A. Because programming languages are made to be used.

They're made to be used by programmers. There would be no point in its existence for it to be somehow published in, you know, like, a book like this, to have documentation made available if the intent wasn't to make it used.

And there's, you know, long history of, like, my entire career of, you know, new programming languages coming along and things get published about them and people use them in various ways.

Q. Again back at the time you were working on the Android team at Google, did you have a view as to whether or not you were free to use declarations for Java APIs?

A. Yeah.

Q. What was your view?

A. I thought that the -- that Java declarations were A-okay to be used.

Q. Why was that?

A. Because I had seen at the time many -- many examples of a programming language coming along published by or built by one set of people, and a different set of people come along and do a new implementation of that language and inevitably you use, you know, the same declarations.

Q. Okay. Do you have a particular example in mind?

A. I'm thinking of the C programming language or C++. Very early in my career I actually had reason to use -- to build a system that was mostly in the C programming language, and we

used three different C compiler and runtime products from three different companies, but we used the same code base. So we used our -- like, our application code was the same. We used three different C implementations to provide the executables, the applications for a few different platforms.

Q. You just described, when you were showing us the sketch you made on the easel, information that you knew back at the time in Android about how much flexibility you did or did not have in how you wrote a declaration.

Back when you were with Google, did that have any effect on your view as to whether or not declarations for the Java APIs were available for use?

A. Okay. That was a bit of a mouthful. Could you --

Q. No problem. I'm happy to --

MS. HURST: Your Honor, I'm going to object. This seems to be about designing APIs. There has been no foundation laid that this witness was in any way involved in that. That was Dr. Bloch's job. We already heard from him.

MS. ANDERSON: Your Honor, we are simply explaining the witness' testimony, and this witness, we will be discussing momentarily, was very much involved in --

THE COURT: Why don't you go at it a slightly different way and tell the story up until the point that he gets to where he's using or redesigning some API, and then you can ask to what extent did you feel you were able to do that;

but you haven't gotten to that part of the story yet.

MS. ANDERSON: I'd be happy to, Your Honor.

THE COURT: All right. So why don't you rearrange it a little bit?

MS. ANDERSON: I'd be happy to do that.

Q. Let's talk about the time you were on the Android team and the work that you did. Okay?

A. Okay.

Q. What were your areas of responsibility working on the Android platform?

A. I was the tech lead responsible for the virtual machine and the core libraries.

Q. Would you please explain to the jury what a virtual machine is?

A. Sure.

Q. Actually let me rephrase. Back when you were with the Android team, what did you understand a virtual machine to be?

A. So at the time I would have understood a virtual machine to be in a way that the engine that runs code in it -- in a particular system. So there's many different kinds of virtual machines. There was a particular one that we were building on Android. I was responsible for that.

Q. You also referenced work you did on the core libraries. Could you be more specific what you worked on at Google on Android relating to the core libraries?

A. I was also the technical lead for the core library work, and that involved a combination of activities. I wrote code myself, and I oversaw other -- other employees who wrote code for the -- for the core libraries. And we also had contractors who were doing work, and we also used some existing open source projects and incorporated them into -- into our work.

Q. To what extent, if at all, did the core libraries you worked on concern implementations for Java APIs?

A. Oh, the core libraries were -- I can't say a hundred percent, but 90-plus percent of the core library work that I'm referring to was implementation of Java APIs.

Q. Okay. And just briefly, if we could take a look at Exhibit 43.1. Just to orient ourselves to what you just said, could you please indicate for the jury where on Exhibit 43.1 -- where the areas you were working on for Android at the Google --

A. Okay. It's actually nicely cleanly separated there. It's the yellowish box, mid right. Yeah.

Q. So are you indicating the yellow box that starts with the words "Android Runtime"?

A. I am.

Q. The testimony you just gave about core libraries, that's the box you're referring to?

A. Yes.

Q. Thank you.

Mr. Dahm, you can take that down.

During what years were you working on implementations for Java APIs for the core libraries while you were at Google?

A. That was ongoing almost from the start. There was definitely a point where we sort of made some, I'll call it, key decisions about how to proceed, and I'll say -- you know, call it full-steam ahead on the core libraries was late 2000 -- starting late 2006 and ongoing from there.

Q. All right. During the time you were working on Android, were you aware of something called GNU Classpath?

A. I was.

Q. What was that?

A. GNU Classpath was an independent implementation of Java core libraries, of Java APIs.

Q. Did any of the work that you did while you were working on Android at Google involve GNU Classpath in any way?

A. It did.

Q. Would you explain how it involved GNU Classpath.

A. Yes. When we were -- when we were first starting out on Android, GNU Classpath was a set of code that we used to -- as part of the Android system.

Q. Okay. And for what purpose was GNU Classpath code used?

A. As an implementation of Java APIs.

Q. Okay. And did the work that you did in connection with GNU Classpath code have any relationship to work you were doing

on the virtual machine?

A. It's -- you know, there's -- there's a relationship in that they had to work together. But there's also, you know, there's -- call it -- there's a lot of separate stuff and then a little bit of sort of gray area in the middle.

Q. How did Google obtain GNU Classpath code during the time you were with the Android team to do the work you were doing?

A. I -- so I wasn't the one who got it. My assumption is that somebody grabbed it off the Internet. It's open source software, freely available.

Q. What other steps, at a high level, did you take in your job responsibilities to work on developing core library implementations for Android?

A. What other steps did I take?

Q. For example, did you do anything with either writing or downloading any other code to create that core library?

A. Yeah. I think I touched on this. So I -- well, in addition to writing code, which I and several of my colleagues did, we also used other open source software. And so we would have -- you know, we pulled that off the Internet. And we also engaged the services of a contractor to do that -- that same kind of thing.

Q. Okay. When you say that you used other open source software, what do you mean?

A. What I mean is that there are -- there are lots -- there's

lots of open source software out there in the world, and some of it happened to be a good fit for what we were trying to do with Android. So when we came across that, one of the options that we had was to import that code into our project and either use it as is or modify it in some way to better suit our environment.

Q. Are you familiar with the phase "Apache Harmony"?

A. I am.

Q. And what is that?

A. Apache Harmony is the -- probably the -- not probably. Safe to say it's the biggest source of core library implementation code that we used in building the -- the Android Runtime.

Q. When you say core library implementation code, are you referring to implementations of Java APIs?

A. I am.

Q. All right. How about Bouncy Castle? Does that phrase mean anything to you?

A. It does.

Q. What is that?

A. That's another open source project that we used as part of our core library implementation.

Q. And, again, are you referring to Java API implementations?

A. I am.

Q. The work that you've been describing that you were doing

on core library implementations while you were at Google, when you refer to implementations, could you identify that in connection with the diagram that you showed us and drew for us a few minutes ago? Where does it fit? What were you working on to create?

A. Okay. What we were doing -- I don't know how easy it's going to be to use the diagram, but what we did was we took the entire -- the entire set of source code, which was a bunch of files, and we brought them into our own directory hierarchy, our own set of source code, and so those files would have looked, you know, roughly like that thing at the bottom right. And so sometimes for some files, it was fine. We didn't have to do anything to it. And sometimes there would be one -- one kind of problem or another, and we would have to modify the file in some way to make it work within the context of Android.

Q. Okay.

THE COURT: It's unclear where in that answer -- where the material came from to begin with.

THE WITNESS: For -- let's say for Apache Harmony, there is an organization called the Apache Foundation. They were the stewards for that project. They run a website. And a set of -- well, they run a set of Internet-available repositories, and the Apache Harmony code was amongst the stuff that was available from the Apache Foundation.

So I or one of my colleagues -- I actually I know I did.

So at one point, I issued some commands on a command line and -- which caused the source code from the Apache Harmony project, hosted at the Apache organization or hosted on their servers -- that came -- so that was -- via network was pulled onto my local computer. From my local computer I issued another series of commands that caused that code to then get integrated into the Android code base.

THE COURT: And so that example came from Apache Harmony?

THE WITNESS: That's right.

THE COURT: And that was, at least as you're telling us -- it was the Java APIs, but with an independent implementation? Is that correct or not correct?

THE WITNESS: Yeah. It was an independent implementation of Java APIs.

THE COURT: Okay. All right. Thank you.

Go ahead.

MS. ANDERSON: Thank you, Your Honor.

Q. You mentioned a minute ago, Mr. Bornstein, that sometimes Google had to work on modifying the open source Apache Harmony code that it had downloaded to work on. Could you give us some examples of why you would have to do that?

A. Sure. Apache Harmony -- I guess -- the thing to remember is any given software project has sort of a context in which it's sort of envisioned to operate. Apache Harmony happened to

be one that was targeted towards servers and desktop computers and not for mobile devices. And there's a lot of differences between a big computer that sits on a desk or on a data center and a little computer that fits on your hand and runs on a battery.

So a lot of what we were doing when we were modifying the Apache code was finding those places where it -- you know, where the code was kind of -- assuming it was in a -- in a different context and changing it -- changing the code to no longer make those assumptions and to in fact make different assumptions.

Q. At the time when you were doing this work, did you have an understanding as to whether the Apache Harmony code was an independent implementation of a particular Java platform?

A. When you -- I'm not sure what you're asking. I'm sorry.

Q. Sure. We've talked a little bit about independent implementations and open source. Did you have an understanding as to which, if any, platform Apache Harmony was an independent implementation of?

A. I would -- speaking very informally, I would have said it was an independent implementation of Java, but there's a lot to unpack in that.

Q. Okay. And you mentioned that Apache Harmony's implementation was more targeted at desktops and servers. How did you know that?

MS. HURST: Objection. Leading.

THE COURT: Well, he did say that, and the follow-up question is not leading. Overruled.

MS. HURST: Your Honor, he said Java SE was more targeted toward desktops. He can't remember whether Apache Harmony was --

THE WITNESS: I did --

MS. ANDERSON: Your Honor, that is exactly what he did not say.

THE COURT: What did you say about -- what was targeted at desktops and servers?

THE WITNESS: I didn't say anything about Java SE. I don't think I had said that term until five seconds ago. I -- what I said --

THE COURT: I thought you said Apache Harmony.

THE WITNESS: I said Apache Harmony.

THE COURT: Was directed at desktops.

THE WITNESS: And servers, yes.

THE COURT: And servers. And that you were modifying it for mobile phones.

THE WITNESS: That's correct.

THE COURT: I think that's what he did say. And you had a "how" question. Repeat the "how" question.

BY MS. ANDERSON:

Q. How did you know that? Why did you believe that about

Apache Harmony?

A. I knew that and I believed that both from seeing the source code as a professional in the industry. Even in 2006 when all this was happening, I had already been in the industry for decades -- decade and a half, at least, and I had been programming since I was seven, so it's like my history with this goes back beyond my professional engagement. I have been doing programming for a very long time, even then, and I could see where this code was making assumptions.

And in addition from the documentation about the Harmony project itself, I knew that that's -- that that was the context that they were thinking about.

Q. Thank you.

During the time you worked on the Android platform, were you familiar with the features of the Android platform in general overall?

A. Yeah.

Q. Did you have a view during the time that you were working on the Android platform as to whether any particular portion of the platform was more important than others?

A. Yeah. Safe to say I had a view.

Q. What was your view?

A. I would have said that the pieces -- it's a cohesive hold. They all had to work together.

Q. Did you think that any one was more important than the

other?
A. No.
Q. Did you consider Java API declarations to be more important than any other part of the Android platform code?
A. No.
Q. Did you know approximately how many lines of code were in the Android software platform when it was released to the public?
A. I do.
Q. And approximately how much -- how many lines of code?
A. It was like 10 or 12 million, something like that.
Q. All right. And in terms of time frame, when was the Android platform fully released in source code form to the public?
A. Well, we had, I'll call it, two major release events. We had what we call a preview release, which was in late 2007, and we had the first product release -- well, we had the first -- the first source code release after the first product release was about a year later.
Q. With respect to the late 2007 release, we have a timeline in the case. We have flagged November 2007, Google releases Android. Is that the date you're referring to?
A. Yeah. That sounds right.
Q. When you say this was a pre-release, could you be more specific --

A. I said preview. I didn't say pre-release.
Q. I apologize. When you say preview release, what did you mean?
A. What I meant was that we intended for the release to be functional enough for developers to get a feel for what Android was about and to be able to write applications on top of Android.
Q. Did the information that was released to the public in November 2007 about Android include information about what Java APIs it would use?
A. It did.
Q. All right. And did that include the Java APIs you worked on as part of the core libraries?
A. It did.
Q. Then with respect to the release you mentioned about a year later --
A. Uh-huh.
Q. -- what was released to the public as part of that release?
A. It was another -- it was a newer version of effectively the same code base. It was the code base plus an additional year's worth of development.
Q. Okay. Did that release include information about what Java APIs were implemented as part of the Android platform?
A. It did.

Q. And did that 2008 release include the actual implementation of the Java APIs that we're talking about?

A. It did.

Q. Can you please tell us approximately how many API packages in total were implemented as part of the Android platform as a whole?

A. How many APIs in Android has a whole?

Q. In Android as a whole, yes.

A. I think it was like 160, 170, something like that.

Q. Were any of those Java API packages?

A. Yeah. About 50 of them.

Q. Okay. Does the phrase "Android name space packages" mean anything?

A. It does.

Q. What does it mean?

A. So if you remember, I was talking about the hierarchy of APIs. So the -- so every -- in a Java System, every class is associated with a package and that package has a name. The Android APIs would be the ones whose name began literally with the word "Android."

Q. Okay. And so about how many Android API packages were among the 165 or 170 total packages?

A. Like 120 or so, something like that.

Q. Thank you.

And who assembled those Android API packages?

A. The Android engineering team as a whole.

Q. Were those the same or different than the Java API packages?

A. They were different.

Q. Why did Google develop Android API packages?

A. Because the short version is there wasn't anything like what we wanted to achieve out there in the world, so that was something that we had to write ourselves.

Q. Could you give us a few examples of goals you wanted to achieve with the Android API packages?

A. Sure. So, for example, we intended to have a smartphone that did multiple -- multiple -- it was able to run multiple applications simultaneously, and there's some -- that has implications on how an application itself is designed, so we had to define Android packages and classes to enable application developers to play in that -- in that world of multiple applications, running on a smartphone-type device.

Q. Thank you.

Did you have any role in determining which Java API packages would be implemented as part of Android's core libraries?

A. I did.

Q. What was your role?

A. As I said before, I was the technical lead for the core libraries, and among the duties that fell to me was figuring

out what made sense to have in our Java APIs.

Q. When you say "figuring out what made sense," what do you mean by that?

A. What I mean is that Android was -- was this different context for running code in general. We were looking at a -- you know, we were looking at a previous, you know -- previously-exiting set of APIs that had been defined. Not all of them made sense in the context of a smartphone, and it was in part up to me to help figure out which pieces of that made sense to sort of -- to include in our core libraries.

Q. And did any of those considerations relate to specific aspects of a mobile device?

A. Yeah.

Q. Can you give us a few more examples of those considerations.

A. Oh, so -- let's see. So there were -- I don't know. I'm sorry. I can't think of, like, a very good example for you.

Q. That's fine. No problem.

Did your team create implementations of all the Java APIs in Java SE?

A. No.

Q. Why is that?

A. It didn't make sense.

Q. Why didn't it make sense?

A. Like I said, I wish I could give you a concrete example.

It's been a while. But in looking through, you know, like what was being defined as part of Java SE, there was -- there was some stuff that just -- you know, just didn't really make sense to have -- actually, here's a good one.

So Java SE included its own idea of what an application looked like. And that implied kinds of interaction models of like what, you know -- what it means to run two applications at the same time, what it means to switch between them, how does an application start up and shut down, and we didn't want to use any of that for Android, so it did not make sense to include that in -- among the APIs that we were using on Android.

Q. All right. Thank you.

Approximately how long did it take, based on your experience while you were at Android, to do the development work required to launch the first full source code stack for Android?

A. So I -- I joined in October 2005. The project was already underway. We released the first phone using it -- "we" in a larger sense -- in, what was it, October of 2008. So -- and I guess before I joined, the project was ongoing in one form or another for something like a year, year and a half. So add that all together.

Q. All right. In or around 2006, did you hear of any announcements by Sun relating to open source?

A. Yes.

Q. And specifically what do you recall hearing coming out of Sun in relation to open sourcing?

A. In late 2006, Sun made an announcement of -- that they were open sourcing an implementation of Java. Yeah. Late -- late 2006.

Q. Okay. At that point, did you and your team at Google stop work on the implementations for the Java APIs and just wait off to see what Sun released?

A. No. There wouldn't have been a point in waiting.

Q. Why is that?

A. An announcement -- an announcement is just an announcement. There's no substance. So they could have announced it and then a month letter they could have decided to unannounce it. So -- and we were working on a product. We were trying to get something out the door. There's no point in just sitting on our thumbs and doing nothing.

Q. Okay. And then did there come a point in the time when you learned that Sun actually did release an open source platform?

A. Yeah. That was several -- several months later or early 2007.

Q. Was that before or after the November 2007 release of Android?

A. It was before the November 2007 release.

Q. Okay. And approximately how many months before that?
A. I think it was like May. Six-ish months, five months, something like that.
Q. At the time that you learned that Sun actually released an open source product, from a technical standpoint, did you make a decision to stop work and just import those open source implementations?
A. We just continued to do -- we did not decide to drop what we were doing and use -- use the Sun code.
Q. Is there any technical reason why you didn't do that?
A. It was too late, if nothing else.
Q. What do you mean by that?
A. So -- okay. So this was -- let's say it was like May of 2007. We had already been working on getting the Harmony-based version of the core libraries out for several months. We were getting -- it was working. It wasn't done, but it was working, and so, you know, you could sort of see a path back in May of 2007 -- you could see a path from there to get to a timely release around November. And that was in comparison to what amounted to an unknown.
Q. And at the time that you heard about the release in '07 of Sun's platform on an open source basis, did you think there was anything wrong with the Google team having created an independent implementation of Java APIs?
A. No.

Q. And again, why is that?

A. Again, because this is a thing that happens in our industry over and over again. This is a very common thing, is that something exists as a product and somebody -- somebody else or some other team takes the ideas of that as a basis and makes a new implementation.

Q. Okay. Thank you.

In your work developing the Android platform, did Google ever consider using a different language for developers for the platform, other than Java?

A. It did.

Q. And what did it consider while you were there?

A. I think the two main -- the two main contenders were a language called C++ and a language called JavaScript, and JavaScript and Java are not the same language.

Q. To what extent, if at all, did you have a view at the time about whether it was a necessity that the Android platform use the Java language?

A. I didn't think it was a necessity at all.

Q. And why is that?

A. It was -- because we had actually gone through a lot of the -- the design exercise of figuring out what it would look like if we used C++, what it would have looked like if we used JavaScript. In fact, we had a prototype that was based on JavaScript that predated my involvement of the project.

Q. All right. Thank you.

Your Honor, I pass the witness.

THE COURT: All right. Thank you. Let's go to cross.

CROSS-EXAMINATION

BY MS. HURST:

Q. Trudy, can I have Exhibit 43.1, please.

Mr. Bornstein, have you seen this before?

A. That's right.

Q. It's part of the documentation that comes from the Android source code?

A. I don't think it ever necessarily came with the Android source code as, like, a bundle. It was certainly used in association with Android at the time.

Q. Well, if your attorneys have represented it's part of the Cupcake source code, do you have any reason to disagree with that?

A. I don't.

Q. Now, is it true that the libraries you've been talking about are these core libraries, the ones that you were in charge of?

A. That was mostly what we've been talking about.

Q. And there were about 50 of those; is that right? Fifty core libraries?

A. Yeah. That's -- the way you're using it, yes.

Q. And 37 of those are the ones that are at issue in this

case?

A. I understand that to be the case, yes.

Q. And everything else above those core libraries depends on those to work; isn't that right?

A. That's a fair statement.

Q. Now, you mentioned that you had GNU Classpath code in your project at one point; is that right?

A. Yes.

Q. And it's true, sir, that you instructed Noser that they absolutely not -- could not use Classpath code because it was covered by an incompatible license.

A. Was there a question there?

Q. Yes. Isn't that true? Isn't it true that you instructed Noser, your contractor, that they could not use the Classpath code because it was covered by an incompatible license?

A. With the understanding that at the time they knew I wasn't a lawyer and I wasn't making a legal judgment, that was my understanding, was that the -- that we did not want to use code as much as possible that used that license.

MS. HURST: Your Honor, I'm handing the witness a document that has been marked Exhibit 9505. This is for impeachment so it has not been previously disclosed.

THE COURT: Is it a document that he wrote?

MS. HURST: It is.

THE COURT: All right.

THE WITNESS: Let's see.

BY MS. HURST:

Q. Mr. Bornstein, would you turn to the last page --

A. Uh-huh.

Q. -- of that exhibit.

A. The last page --

Q. Page 3 of 3.

A. Page 3 of 3. Okay.

Q. And do you see there, sir, there's an exchange on May 18th, 2007, at 6:34 p.m. between you and a Daniel Diaz?

A. That's right.

MS. HURST: Move to admit, Your Honor.

THE COURT: Is this written by him?

MS. HURST: Yes.

THE COURT: Any objection?

MS. ANDERSON: Your Honor, it's not impeachment and it wasn't disclosed. The witness hasn't contradicted anything in this document.

THE COURT: Well, I don't know if it impeaches or not. Why don't you establish the impeachment part, and then the document itself doesn't come into evidence, but maybe the impeaching part does. Maybe we'll put the whole thing in. Why don't you go ahead and you can verbalize the parts that you think impeach what he said.

MS. HURST: Thank you, Your Honor.

Q. Is it true, Mr. Bornstein, that in this email you wrote, "Besides Harmony, is it" -- pardon me. Mr. Diaz wrote to you -- let me the start over.

Is it true that Mr. Diaz wrote to you, "Besides Harmony, is it allowed to copy the Java docs from Classpath?"

And you answered, "Absolutely not. The Classpath code, including the comments, is covered by an incompatible license."

That was your exchange, wasn't it?

A. I think you misquoted Mr. Diaz.

Q. Let me read it again.

A. Okay.

Q. Did Mr. Diaz write to you, "Besides Harmony, it is allow copy the Java docs from Classpath," question mark? Did I read that exactly write?

A. Minus the single quotes, but, yes.

Q. Let me try again.

"Besides Harmony, is it" -- "it is allow," single quote, "copy," single quote, "the Java docs from Classpath," question mark. Did I get it right that time?

A. Yeah. I think so.

Q. And your answer, could you read it for us, sir?

A. Sure.

"Absolutely not. The Classpath code, including the comments, is covered by an incompatible license."

Q. And that was your instruction to Noser, wasn't it?

A. That was an instruction to Noser, that's right.

MS. HURST: Move to admit the document.

THE COURT: It will come into evidence. What's the number?

MS. HURST: 9505.

(Trial Exhibit 9505 received in evidence)

BY MS. HURST:

Q. Classpath, that was exactly the same license that was on Java SE when it was released; isn't that true?

A. I don't know that for sure.

Q. And isn't it true, sir, that you didn't use the OpenJDK source code because the license was incompatible with Android's needs?

A. I think the more salient thing is that it was too late.

Q. So you're telling me here and this jury and the judge under oath here today that it was about timing and not the license; is that right?

A. I'm not going to say it was a hundred percent either one. You know, a lot of factors go into any decision.

Q. Isn't it true, sir, that it was not about the timing? And instead, it was about the incompatibility of the license?

A. If I -- sitting here today, remembering what I remember, I think it was much more that we were -- you know, like the train -- the Harmony train had left the station. We were well on our way to getting it done. You know, technically we could

have dropped it, but it's like -- there wouldn't have been a point to it at that point, I don't think.

MS. HURST: Your Honor, may I approach the witness with Exhibit 5048?

THE COURT: Yes, you may.

BY MS. HURST:

Q. Mr. Bornstein, do you recognize that as an email you wrote to Ramy Dodin? Did I pronounce that correctly?

A. I don't think you did. I think he pronounces his name Ramy.

Q. How do you pronounce the last name?

A. I think you've got that right.

Q. Is that an email you wrote to Mr. Dodin?

A. It looks like it.

Q. On November 28, 2006, late November, 2006?

A. That's right.

Q. That was the period of time that you were talking about when Ms. Anderson was asking you questions on direct; right?

A. We also talked about around May when Sun actually released the code.

Q. But it was late November when you said you were full-steam ahead; isn't that right? That's what you said on direct?

MS. ANDERSON: Objection. Mischaracterizes testimony and is argumentive.

THE COURT: I don't remember if that's what he said,

but is that what you said on direct?

THE WITNESS: I don't believe that's what I said. Can I clarify?

MS. HURST: I'll continue, Your Honor.

THE COURT: Go ahead. But --

MS. HURST: I move to admit Exhibit 5048.

MS. ANDERSON: No objection.

THE COURT: Thank you. Received.

(Trial Exhibit 5048 received in evidence)

MS. HURST: All right.

Q. So this is an email exchange between you and Mr. Dodin; is that right?

A. Yes.

Q. "There is a question I forgot to ask you the other day. Will Sun's recent announcement about open sourcing Java happen soon enough to benefit Android?" Do you see that?

A. I do.

Q. And that was from Dr. Dodin, right, the little carrot, that's his question?

A. I believe so.

Q. Right?

A. Uh-huh.

Q. And your answer -- sir, could you just read that answer for us, please.

A. Okay.

"It's not about timing so much as details. The licensing that Sun is using for both SE and ME are incompatible with Android's needs. I'm happy to talk further about it in person."

Q. And that was what you wrote before there was any lawsuit 'isn't that true?

A. That's right.

Q. That was the same license that was on Classpath that you also instructed Noser not to use; isn't that true?

A. Well, at the time --

Q. "Yes" or "no"?

MS. ANDERSON: Objection, Your Honor.

THE COURT: Well, it's a fair question. You can say "yes" or "no" and then give your explanation, but was it the same license?

THE WITNESS: I don't know if it was the same license. It did have a similar name, but I don't remember reading either one of those licenses and comparing them side by side, so I can't tell you truthfully that they were the same license.

THE COURT: Could you have told somebody truthfully back then?

THE WITNESS: Not in late 2006 because it hadn't been released. We only -- we only had -- we only knew kind of hearsay what we expected it to be licensed as, not as -- there was no actual open source release. Remember, there was an

announcement and then later there was a release, and so in that intermediate time, there was a lot of questions about what were they actually going to do. Again, at this point --

THE COURT: All right. All right. Okay. Next question. Thank you. Next question.

BY MS. HURST:

Q. Let's look at that 9505. Do you have that still?

A. Yeah.

Q. That's May 18, 2007; right?

A. That seems to be the date at the top.

Q. That's ten days after OpenJDK was released, isn't it?

A. I don't remember the exact date of the OpenJDK release.

Q. It was ten days after OpenJDK released that you wrote, "Can you copy the Java docs from Class" -- pardon me.

"Absolutely not. The Classpath code, including the comments, is covered by an incompatible license."

You understood that then; right?

A. Yeah.

Q. Let me show you Exhibit 415.

A. Uh-huh.

Q. Is that an email that you wrote to Ray Chen dated February 26, 2007?

A. It looks like it.

MS. HURST: Move to admit, Your Honor.

MS. ANDERSON: No objection, Your Honor.

THE COURT: All right. 415 is now in evidence.

(Trial Exhibit 415 received in evidence)

BY MS. HURST:

Q. All right. And, Mr. Bornstein, let's look here at the paragraph starting with the words "we thought."

A. Uh-huh.

Q. Now, were you kind of involved in open source at Google?

A. It depends on what you mean.

Q. Well, did -- was it something that you considered or thought about in any way?

A. Well, in my role, I had to be aware of open source projects and I had to take activity that was related in one way or another to open source.

Q. And you made instructions like you did to Noser about which open source projects were okay to use and which ones weren't; right?

A. I would say --

Q. "Yes" or "no." Did you make instructions, as you did to Noser, about which open source projects were okay to use and which ones weren't?

A. I was a point of contact and I communicated consensus.

Q. All right.

A. I don't know if that's what you're asking about.

Q. You communicated consensus like this consensus referred to in Exhibit 415.

"On the licensing front although Classpath would probably be okay in my opinion, the lawyer-advised consensus is that there is a potential for trouble. The same situation holds with Sun's release of their own class libraries under a very similar license."

That's the consensus that you communicated to Mr. Chen in Exhibit 415; is that right, Mr. Bornstein?

A. That's a fair statement.

Q. And, in fact, you thought the Classpath exception license was an attractive nuisance licensing nightmare waiting to happen; isn't that true?

A. I believe -- that sounds like my words. I can believe I said it.

Q. And your interests with regards to the licensing question was to steer well clear of any potential problems. "Get your product shipped and have potential customers, i.e., other companies with paranoid legal staffs, not to have to worry about using what you'd done"' isn't that true?

A. I think -- again, that sounds like you're reading words that I wrote.

Q. Yeah. That sounds like you, doesn't it?

A. I think I'm usually a little more pleasant how I convey it.

Q. Mr. Bornstein, have you ever heard of Simon Phipps?

A. I don't think so.

Q. Did you ever hear of a guy who was a chief open source officer at Sun?
A. I'm not surprised that that job title exists, but other than that, I don't know.
Q. When selecting which Java API packages to use out of Java SE, is it true that your goal was "to achieve approximate parity with CDC personal basis profile which has most of the classes from SE that one could reasonably expect to be useful on a high-end mobile device of resent or soon to be vintage"; is that true?
A. That again sounds like you're quoting something that I wrote so I would say at the time I wrote it, that was probably an accurate representation of my beliefs.
Q. Now, Mr. Bornstein, did there come a time when you decided to scrub the source code of Android for various terms?
A. Yeah.
Q. And you decided to scrub the source code of what you called the J word, which meant Java; is that true?
A. Yeah. It was kind of like a joke.
Q. Well, joke or not, was it your understanding, sir, that you didn't own the Java API and couldn't go around altering it?
A. Yeah. That's because nobody owned the API.
Q. I will show you Exhibit 234, Mr. Bornstein.
A. Okie doke.
Q. Is that an email exchange?

A. Yes, it's an email exchange.

Q. Involving you dated April 6, 2009?

A. So I see that is the date. I see my name in the quotes there so, that's a fair statement.

MS. HURST: Move to admit Exhibit 234, Your Honor.

MS. ANDERSON: No objection, Your Honor.

THE COURT: All right. Received.

(Trial Exhibit 234 received in evidence)

THE COURT: We are down to about three minutes.

MS. HURST: All right. I'm going to go fast and get it down.

Q. Mr. Bornstein, in this email, did you write, "Bob is right. We don't, quote/unquote, own the java.star API and so can't go around altering it. The party line is that we would love to pass the TCK for it if only we could legally get access to it"? Did you write that, sir?

A. I did.

Q. All right. Let me show you Exhibit 348. Do you recognize Exhibit 348 as the complete list of words that you scrubbed from the Android source code?

A. I don't know that I ever have seen this actual email before.

Q. Well, do you recognize it as the list of words that you sought to scrub from the Android source code?

A. I think there's -- I don't remember reading this

particular document. Those names look like a reasonable representation of the sorts of things that we were looking for.

MS. HURST: Move to admit Exhibit 348, Your Honor.

MS. ANDERSON: Objection, Your Honor. The exhibit itself lacks foundation and is hearsay, but the witness can certainly be asked about specific words.

MS. HURST: Your Honor, there is an agreement about this exhibit, because Mr. Morrill is sick and unavailable to attend trial, that his documents would be admitted.

MS. ANDERSON: I'm not aware of such an agreement.

THE COURT: Who do you have your agreement with?

MS. HURST: I believe that was agreed to between Mr. Bicks and Mr. Van Nest.

MR. KAMPER: Your Honor, the agreement is that because Mr. Morrill is -- just had surgery, that he would be played by video by Oracle in their case, and at that point, they will be able to admit exhibits related to him.

MS. HURST: Your Honor, subject to that video being played, I would like to display the document and examine the witness.

THE COURT: We only have two minutes.

MS. HURST: Two minutes, that's all it's going to take.

THE COURT: Let me see the document.

MS. HURST: Your Honor, we're not going to look at the

top part. It's the middle part where it says "included for obvious reasons."

THE COURT: Do you represent to me that this will get into evidence otherwise?

MS. HURST: Yes, Your Honor.

THE COURT: Based on that representation, I will allow it in *de bene*, D-E B-E-N-E, and you may examine on this and show it to the jury.

MS. HURST: Thank you, Your Honor.

(Trial Exhibit 348 received in evidence)

BY MS. HURST:

Q. Now, Mr. Bornstein, is it true that you scrubbed the Android code for the words "Sun/Oracle, "Oracle/Java" and the word "license"?

A. Do you mean me personally or do you mean us as a team?

Q. You at Google.

A. Okay. Me, personally --

Q. You, Google the entity.

A. Oh, oh, oh, sorry.

MS. ANDERSON: Objection, Your Honor. That question fundamentally lacks foundation.

THE COURT: To the extent you know one way or the other what Google did on scrubbing, you should answer the question, even if you didn't personally do it.

THE WITNESS: Well, so I in fact did some scrubbing

myself. I don't remember the exact terms that ended up being the ones that I did, but I know that I was aware of the term "Java," for example.

BY MS. HURST:

Q. So you knew you scrubbed the term "Java"?

A. Yeah.

Q. And did you scrub the term "Oracle"?

A. You know, I don't remember seeing "Oracle."

Q. Did you scrub the term "Sun"?

A. I don't -- I don't a hundred percent remember seeing the term "Sun" either.

Q. Did you scrub the term "license"?

A. I don't specifically remember that one either.

MS. HURST: No further questions.

THE COURT: All right. We are going to break, unless you can do it in one minute.

MS. ANDERSON: Your Honor, I can't do it in one minute.

THE COURT: Then we are going to end for the day.

I want to thank the jury for all of your close attention. I know this is a difficult course for you to take, but it's an interesting one, so good for you.

Please have a great weekend. Don't do any research about the case. Don't talk with anyone about the case. Don't read news accounts about the case.

Have a great weekend. We'll see you back here on Monday morning at the normal time.

(Proceedings were heard out of presence of the jury:)

THE COURT: Everyone be seated.

Mr. Bornstein, you've got to be back at 7:30 on Monday. Okay?

THE WITNESS: Okay.

THE COURT: You can step out now, unless we need the witness for anything. Do we?

MS. ANDERSON: No, Your Honor.

MS. HURST: No, Your Honor.

THE COURT: Okay. Anything that the lawyers need me for?

MR. VAN NEST: Yes, Your Honor.

THE COURT: What's that?

MR. VAN NEST: We have one more deposition excerpt to hand up for Your Honor. It's been put together by the parties.

THE COURT: I'll look at it over the weekend. And anything else?

MR. VAN NEST: There's -- it's just been called to my attention, you asked the other day about the press and the press asking you for material.

THE COURT: I mentioned it right this morning. I put out an order asking you all to help the press get some documents.

MR. VAN NEST: It's just been called to my attention that some of these document clips or maybe video clips are being published on Twitter, maybe before they're shown in court. I'm not sure what's going on. But this just showed up this morning. And it looks like somebody is getting this stuff out to the press before -- maybe even before they're played here in court.

So before anything else happens, I would just like to get some guidance from Your Honor on what is appropriate --

THE COURT: Are we releasing anything to the press ahead of time?

THE CLERK: No, Your Honor.

THE COURT: We haven't done it. How about over there? Are you putting it out, giving it --

MR. BICKS: No.

MS. HURST: No.

THE COURT: I don't know how the leaking is occurring. What do you want me to do about it, Mr. Van Nest?

MR. KAMBER: Your Honor, I believe that somebody on the Oracle side may have given a disk or a thumb drive with exhibits to somebody in the press gallery.

MS. HURST: That's what the order said, Your Honor. We got it off the ECF and complied with it immediately.

THE COURT: Is it all admitted material?

MR. KAMBER: No. They are pre-releasing all of the

AEO-designated documents that are potential trial exhibits before they're actually shown --

MS. HURST: No. The instructions were to only use the admitted exhibits, Your Honor. That was my instruction to our team, and if that was not followed correctly, I apologize. I'll find out immediately.

THE COURT: You find out and report back. Yes. Only admitted material.

Here are the minutes that you've used up. Very close. Oracle has used 458 minutes. Google has used 457 minutes. One minute apart. But you're both over the halfway mark now.

MS. HURST: Your Honor, we're confident it was only the admitted materials that we provided.

THE COURT: Thank you. All right. Look, I'm not going to solve every problem if the world. Just obey the orders of the Court, please.

Anything else for today?

MR. BICKS: No, Your Honor.

MR. VAN NEST: I don't believe so, Your Honor. No. See you Monday.

THE COURT: Thank you. Have a good weekend.

(Proceedings adjourned at 1:04 p.m.)

CERTIFICATE OF REPORTER

I certify that the foregoing is a correct transcript from the record of proceedings in the above-entitled matter.

DATE: Friday, May 13, 2016

Pamela A. Batalo

Pamela A. Batalo, CSR No. 3593, RMR, FCRR
U.S. Court Reporter